

# **TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: X2612 – Elektrotechnika a informatika

Studijní obor: 1234T567 – Název studijního oboru

## **Program pro výuku FFT analýzy**

## **Program for FFT analysis education**

### **Diplomová práce**

Autor:	<b>Pavel Vaněk</b>
Vedoucí práce:	Doc. Ing. Ivan Jaksch CSc.
Konzultant:	Ing. Petr Fuchs
	Ing. Petr Pelant, Škoda auto a.s.

**V Liberci 10. 5. 2008**

/\*Zadání\*/

## Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Na tomto místě bych chtěl poděkovat vedoucímu mé diplomové práce Doc. Ing. Ivanu Jakschovi CSc. za odbornou pomoc, rady a čas, který věnoval mé práci. Dále bych rád poděkoval Ing. Petru Pelantovi ze společnosti Škoda auto a.s. za konzultace a přípravu zadání diplomové práce. Mé poděkování patří rovněž celé mé rodině a všem mým blízkým za velkou podporu během celé doby mého studia.

## **Anotace**

Diplomová práce se zabývá výukou analýzy signálů a jejím výsledkem je program, který interaktivní formou seznámí uživatele s analýzou signálů pomocí rychlé Fourierovy transformace (FFT) a analýzou s konstantní relativní šířkou pásma (CPB). Program předvede jednak analýzu (dekompozici) signálu rozkladem na jednotlivé harmonické složky, tak i syntézu (skládání) jednotlivých harmonických složek. Skládání je graficky vyřešeno jednak překládáním jednotlivých sinusových průběhů přes sebe, tak i postupnou deformací základního sinusového průběhu. Pro získání časových průběhů signálů, které se budou analyzovat, obsahuje program vlastní generátor signálů (sinus, obdélník, trojúhelník, šum,...). Dále je možné signál získat nahráním pomocí zvukové karty nebo načtením signálu ze souboru ve formátu wav. Zobrazovaný signál je samozřejmě možné pomocí zvukové také karty přehrávat. Základní obrazovka programu nabízí pohled na časový průběh signálu a na jeho frekvenční spektrum. Zobrazený časový průběh signálu je možné modifikovat a sledovat, jak se tyto změny projeví ve frekvenčním spektru. A naopak zobrazené frekvenční spektrum jde rovněž upravovat změnou amplitud nebo fází jednotlivých složek a sledovat, jak se tyto změny projeví na časovém průběhu signálu. Program dále ukazuje některé další metody pro měření a vyhodnocování spekter signálů, jako je použití různých okénkovacích funkcí a jejich vliv na spektrum signálu.

Součástí programu je také nápověda, která obsahuje úvod do analýzy signálů, teorii Fourierovy transformace a její historii, stručný popis CPB analýzy a způsoby její syntézy.

## **Klíčová slova:**

Signály, analýza signálů, výuka, Fourierova transformace, CPB

## **Annotation**

Diploma Thesis deals with signals analysis and its result is a program, which interactively familiarizes user with signals analysis by using Fast Fourier Transformation (FFT) and Constant Percentage Bandwidth analysis (CPB). Program will demonstrate signal analysis (decomposition) by fragmentation to separate harmonic elements and synthesis (composition) of single harmonic elements as well.

Composition is graphically solved both by translantioning of single sine curves crisscross and step by step deformation of the basic waveform. For receiving the waveforms, which are being analyzed, the program has own generator of the signals (sine, rectangle, triangle, acoustic noise,). Next is possible to get the signal recording by sound card or to load the signal from wav data format file. That signal is of course playable by sound card. Basic program screen offers look at waveform and its frequency spectrum. The visible waveform is possible to upgrade and observe how the changes will show itself in frequency spectrum. And in other way the frequency spectrum is also possible to arrange, by changing amplitudes or phases of simple components and see, how these changes will express in waveform. And furthermore program shows some other methods for measurement and solving spectrum signals such as usage of various window functions and their influence on a signal spectrum

A part of the program is also help, which contain guide to analysis of signals, theory of Fast Fourier transform and its history, brief characterization of CPB analysis and ways of its synthesis.

## **Key words:**

Signals, signal analysis, Fourier Transformation, CPB

# Obsah

SEZNAM POUŽITÝCH SYMBOLŮ .....	10
SEZNAM ZKRATEK .....	10
ÚVOD .....	11
<b>2 ÚVOD DO ZPRACOVÁNÍ SIGNÁLŮ .....</b>	<b>12</b>
2.1 SIGNÁLY .....	12
2.2 DĚLENÍ SIGNÁLŮ .....	12
2.2.1 <i>Deterministické signály</i> .....	13
2.2.2 <i>Náhodné signály</i> .....	13
2.3 FOURIEROVA TRANSFORMACE .....	14
2.4 FOURIEROVY ŘADY (FOURIERŮV ROZVOJ) .....	14
2.5 FOURIEROVA TRANSFORMACE OBECNÉHO SIGNÁLU .....	15
2.5.1 <i>Fourierova transformace diskrétního signálu (DTFT)</i> .....	15
2.5.2 <i>Diskrétní Fourierova transformace</i> .....	16
2.6 RYCHLÁ FOURIEROVA TRANSFORMACE FFT .....	16
2.6.1 <i>Odvození FFT</i> .....	17
2.7 ALIASING .....	18
<b>3 VYHODNOCOVÁNÍ SIGNÁLŮ .....</b>	<b>21</b>
3.1 VÝPOČET AMPLITUDOVÉHO A FÁZOVÉHO SPEKTRA SIGNÁLU .....	23
3.1.1 <i>Výkonové spektrum</i> .....	23
3.1.2 <i>Efektivní hodnota amplitudy a fáze ve stupních</i> .....	23
3.1.3 <i>Převod do logaritmických jednotek</i> .....	24
3.1.4 <i>Zobrazení výsledků v decibelovém měřítku</i> .....	25
3.1.5 <i>Odhad výkonu a frekvence</i> .....	26
3.2 OKTÁVOVÉ KMITOČTOVÉ PÁSMO .....	26
3.3 TŘETINOOKTÁVOVÉ PÁSMO .....	28
3.4 ZPŮSOBY VÝPOČTU CPB ANALÝZY .....	29
3.4.1 <i>CPB syntézou z FFT</i> .....	29
3.4.2 <i>CPB vypočtené kaskádní FFT</i> .....	30
3.4.3 <i>CPB vypočtené pomocí skupiny filtrů</i> .....	31
3.4.4 <i>Metoda použitá v programu</i> .....	32
<b>4 ROZPTYL SPEKTRA .....</b>	<b>33</b>
4.1.1 <i>Signál s celočíselným počtem period</i> .....	35
4.1.2 <i>Signál s neceločíselným počtem period</i> .....	36
4.2 OKÉNKOVACÍ FUNKCE .....	38

4.3	VLASTNOSTI OKÉNEK .....	42
4.3.1	Šířka pásma šumu .....	43
4.4	PŘEHLED TYPŮ OKÉNEK .....	44
4.4.1	Obdélníkové (implicitní) okénko .....	44
4.4.2	Hanningovo okénko .....	45
4.4.3	Hamming okénko .....	45
4.4.4	Okénko Kaiser-Bessel .....	46
4.4.5	Trojúhelníkové okno .....	47
4.4.6	Flat Top .....	48
4.4.7	Exponenciální okénko .....	49
4.4.8	Okénko Exact Blackman .....	49
4.4.9	Blackman, Blackman-Harris, Blackman-Nuttall .....	50
4.4.10	Gaussovo okénko .....	51
4.4.11	Okénko Force .....	51
4.4.12	Další typy okének .....	52
4.5	STRATEGIE PRO VÝBĚR OKÉNEK .....	52
<b>5</b>	<b>ZVUKOVÝ VSTUP A VÝSTUP .....</b>	<b>54</b>
5.1	OVĚŘENÍ VLASTNOSTÍ ZVUKOVÉ KARTY .....	54
5.1.1	Co je RMAA a jak pracuje .....	54
5.1.2	Co program naměří .....	55
5.2	API FUNKCE PRO OVLÁDÁNÍ MULTIMÉDIÍ VE WINDOWS .....	56
5.2.1	Co to je API? .....	56
5.3	PRÁCE S MULTIMÉDIÍ .....	56
<b>6</b>	<b>PŘEHLED FORMÁTU WAVE .....</b>	<b>58</b>
6.1	FORMÁT DAT .....	58
6.1.1	Řetězce .....	58
6.2	STRUKTURA SOUBORU .....	59
6.3	HLAVIČKA WAVE SOUBORU .....	60
6.3.1	Blok RIFF .....	60
6.4	BLOKY WAVE SOUBORU .....	61
6.4.1	Blok Formát - "fmt" .....	61
6.4.2	Blok Data - "data" .....	63
6.4.3	Další možné bloky wave souborů .....	64
6.5	ODCHYLKY OD FORMÁTU .....	64
<b>7</b>	<b>POPIS VLASTNÍ APLIKACE .....</b>	<b>66</b>
7.1	ČLENĚNÍ ZDROJOVÉHO KÓDU PROGRAMU .....	66



7.2	UŽIVATELSKÉ ROZHRANÍ PROGRAMU .....	68
7.2.1	<i>Hlavní panel aplikace</i> .....	68
7.2.2	<i>Panel generátor</i> .....	71
7.2.3	<i>Panel nastavení</i> .....	71
7.2.4	<i>Nápověda programu</i> .....	72
7.3	PRAKTICKÉ OVĚŘENÍ PROGRAMU .....	74
<b>8</b>	<b>ZÁVĚR</b> .....	<b>75</b>
<b>9</b>	<b>CITOVANÁ LITERATURA</b> .....	<b>76</b>

## Seznam použitých symbolů

$g(t)$	Obecná funkce času
$x(t)$	Obecný spojitý vstupní signál
$x[n]$	Obecný diskrétní vstupní signál
$y(t)$	Obecný spojitý výstupní signál
$y[n]$	Obecný diskrétní výstupní signál
$A$	Amplituda harmonického signálu
$\omega$	Úhlová frekvence harmonického signálu
$\Delta t$	Vzorkovací perioda
$f_s$	Vzorkovací frekvence
$T$	Perioda signálu, doba záznamu
$\Delta f$	Frekvenční krok
$\delta(t)$	Diracův puls
$\eta(t)$	Jednotkový skok
$F\{\}$	Fourierova transformace
$H()$	Obecný obraz transformace
$G(j\omega)$	Frekvenční spektrum obdélníkového okna
$F_k$	Komplexní koeficienty Fourierovy řady
$X_k$	Komplexní koeficient DFT
$f_d$	Diskrétní frekvence
$B$	Decimační faktor
$f_c$	Centrální frekvence pásma
$D$	Decimace
$B$	Šířka pásma

## Seznam zkratk

DFT	Diskrétní Fourierova transformace
FFT	Rychlá Fourierova transformace
FT	Fourierova transformace
FŘ	Fourierovy řady
Imag	Imaginární část Fourierovy transformace nebo analytického signálu
Real	Reálná část Fourierovy transformace nebo analytického signálu
PSD	Výkonová spektrální hustota
PWR	Výkon
RMS	Efektivní hodnoty
S/N	Odstup signálu od šumu

## Úvod

Teorie zpracování signálu byla dříve rozvíjena téměř výhradně v rámci sdělovací techniky. Kdy bylo možné frekvenční spektrum měřit jen přeladitelným filtrem, nebo pomocí malé skupiny pevně naladěných filtrů s omezenou selektivitou. V posledních letech však tato teorie nachází uplatnění i v dalších oblastech, jako je například strojírenství nebo automobilový průmysl. Zejména pak roste její význam při analýze dynamiky strojů pomocí měření hluku a vibrací, protože tyto signály obsahují informace o technickém stavu zařízení. Podle signálu se usuzuje na dynamické namáhání strojů a jejich technický stav. Zjištění těchto skutečností nám pomůže předpovědět spolehlivost a životnosti zařízení, případně příčinu jeho poruchy.

Tento rozvoj analýzy signálů, byl zapříčiněn zejména prudkým vývojem digitální techniky, zejména pak signálových procesorů a rychlých A/D převodníků. Dále pak a objev J. W. Cooleye a J. W. Turkeye z roku 1965, který dovoľoval podstatně urychlit výpočet Fourierovy transformace z časového záznamu. To způsobilo, že měření časových signálů se stalo jednoduchou technickou úlohou. Proto již není problém pořídit a zpracovat záznamy o desítkách tisíc vzorků. Lze předpokládat, že se stále více techniků setká s vyhodnocením těchto záznamů nejen na oddělení výzkumu a vývoje, ale také i v dalších oblastech jako je kontrola jakosti nebo údržba. Tyto lidé nebudou algoritmy pro zpracování signálu programovat. Budou zadávat parametry výpočtů těmto programům nebo budou připravovat FFT analyzátory. Proto je důležité, aby se předešlo chybám, které lze snadno udělat. Z těchto důvodů vznikl ve společnosti Škoda Auto a. s. požadavek na výukový program, který seznámí uživatele základy analýzy signálů a s algoritmem rychlé Fourierovy transformace

Diplomová práce je zaměřena na vysvětlení základů analýzy signálů a předcházení možným chybám měření. Tyto chyby plynou z fyzikálních omezení, ze kterých se měření a vyhodnocování signálu provádí. K chybám měření dochází, při nestacionaritě měřeného signálu nebo při špatné volbě délky záznamu frekvenčního a dynamického rozsahu vstupů A/D převodníku atd. V práci je popsáno, jak některým těmto chybám předcházet. Je zde zdůvodněna volba nastavení parametrů signálového analyzátoru.

## 2 Úvod do zpracování signálů

Teorie signálů a jejich zpracování hraje podstatnou roli v řadě oborů lidské činnosti. Na jejich základech stavějí vědecké disciplíny v oboru sdělovací techniky, radiotechniky, akustiky, seismologie, diagnostiky, energetiky, počítačového zpracování řeči hudby a obrazu, a jistě mnoha dalších. Přestože tyto aplikace vedou do výrazně odlišných oborů, existuje společný základ, který je spojuje. Tímto základem je analýza signálů a právě její výukou se tato práce zabývá.

### 2.1 Signály

Signály jsou funkce jedné nebo více nezávislých proměnných (většinou funkce času) a nesou informace o podstatě a vlastnostech svého zdroje (Uhlíř, 1995). Nebo mohou obsahovat informace do signálu záměrně zakódované. Signály mohou nést informace nejrozličnějšího charakteru. Informace je v signálu reprezentována časovými změnami okamžité hodnoty fyzikální veličiny, kterou signál reprezentuje. Příkladem takového signálu jsou změny akustického tlaku vzduchu. Naše uši pak takovéto změny zaznamenávají a tím nám umožňují slyšet. Přestože při popisu signálů nemusí být nezávislou proměnnou pouze čas. Pro zjednodušení budeme v dalších částech práce považovat za nezávislou proměnnou vždy čas.

### 2.2 Dělení signálů

Signály můžeme rozdělit do dvou základních skupin. První skupinou jsou signály spojité, někdy nazývané analogové. Jsou to signály, jejichž funkční hodnota je definovaná pro všechny hodnoty nezávisle proměnné. Druhou skupinou jsou signály diskrétní. Abychom tyto dvě skupiny signálu odlišili, budeme používat pro označení spojité nezávisle proměnné symbol  $t$ , kde  $t \in \mathbb{R}$ . A symbol  $n \in \mathbb{Z}$  pro označení diskrétní proměnné. Dále můžeme signály rozdělit na deterministické a náhodné.

- **Deterministické** signály můžeme zapsat vztahem, rovnicí nebo nerovností. Pro každý čas  $t$  či  $n$  víme, jaké hodnoty signál nabude.
- **Náhodné** signály popsat rovnicí nemůžeme a pro čas  $t$  resp.  $n$  nikdy přesně nevíme, jaká bude jejich hodnota. Můžeme je charakterizovat pouze pomocí parametrů, jako jsou např. střední hodnota nebo rozptyl.

### 2.2.1 Deterministické signály

Deterministické signály dále dělíme podle jejich periodicity do dalších čtyř skupin.

- **Periodické signály**, ty jsou složeny z harmonických signálů o frekvencích, které jsou celistvým násobkem jedné základní frekvence.
- **Kvaziperiodické signály** jsou složeny z harmonických signálů o frekvencích, které jsou násobky nejméně dvou základních frekvencí a současně jsou v poměru určeném iracionálním číslem.
- **Přechodné signály** mají nenulovou část na rozdíl od předchozích kategorií časově omezenou. Jedná se např. o přechodné děje, odezvy na impulzní vybuzení apod.
- **Pseudonáhodné** signály jsou periodické signály, vytvořeny tak, aby se svými statistickými vlastnostmi podobaly signálům náhodným. Perioda musí být tak velká, aby se z pohledu konkrétní délky signálu jevíly jako stochastické.

### 2.2.2 Náhodné signály

Náhodné signály někdy také nazývané jako stochastické signály dělíme podle průběhu jejich vlastností v čase do následujících třech skupin.

- **Stacionární signály** mají statistické vlastnosti neměnné v čase. Můžeme je ještě rozdělit na další podskupiny. Signály **stacionární v širším smyslu** nebo také slabě stacionární (wide-sense stationary). Mají střední hodnotu a disperzi nezávislou na čase a autokorelační funkce nezávisí na počátku záznamu signálu. U signálů stacionární v **užším smyslu** (silně stacionární). Kromě předchozích podmínek nezávisí na čase také hustoty pravděpodobnosti všech řádů. **Ergodické** signály mají statistické parametry vypočtené z jednoho úseku signálu, shodné s parametry vypočtenými ze souboru mnoha úseků.
- **Nestacionární** – jejich statistické vlastnosti se v průběhu času mění. Hranice mezi nestacionárními a stacionárními signály je nutno určovat s přihlédnutím ke zkoumané délce signálu.
- **Cyklostacionární signály** jsou signály, jejichž statistické vlastnosti se v průběhu času cyklicky mění.

## 2.3 Fourierova transformace

Jean Baptiste Joseph Fourier (21. 3. 1768 – 15. 5. 1830) byl významný francouzský matematik a fyzik, který ve své práci „Theorie analytique de la chaleur“ (Analytická teorie tepla, 1822), položil základy harmonické analýzy. S touto myšlenkou harmonické analýzy však Fourier nepřišel jako první. Již před ním se jí zabývalo několik slavných matematiků. Jedním z nich byl například Leonard Euler. I když Fourierovy řady nejsou výhradně Fourierovým objevem, nesou jeho jméno. Protože on byl první, kdo ukázal, že jsou silným matematickým nástrojem v matematické fyzice i v matematické analýze.



## 2.4 Fourierovy řady (Fourierův rozvoj)

Fourierovy řady dále jen FŘ, podle (Oppenheim, 1989) umožňují libovolný periodický signál  $x(t)=x(t+iT)$ , kde  $T$  je jeho perioda a  $i=\pm 1, \pm 2, \pm 3 \dots$  je její násobek rozložit na jednotlivé harmonické složky. Tyto složky jsou sinusoidy, jejichž frekvence je celočíselným násobkem frekvence signálu. S Fourierovým rozvojem (FŘ) se můžeme setkat ve třech různých tvarech.

### a) Trigonometrický tvar

$$x(t) = a_0 + \sum_{k=0}^{\infty} [a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)] \quad (2.1)$$

*Nevýhodou tohoto vztahu jsou tři druhy koeficientů.*

### b) Polární tvar

$$x(t) = \sum_{k=0}^{\infty} c_k \cos(k\omega_0 t + \varphi_k) \quad (2.2)$$

*Kde  $c_k$  je amplituda  $k$ -té složky a  $\varphi_k$  je její fáze.*

### c) Nejrozšířenější je exponenciální tvar

$$x(t) = \sum_{k=-\infty}^{\infty} X_k \exp(jk\omega_0 t) \quad (2.3)$$

$$X_k = \int_0^T x(t) \exp(-jk\omega_0 t) dt \quad (2.4)$$

*Pro reálný signál jsou koeficienty  $X_k$  a  $X_{-k}$  jsou komplexně sdružené.*

Kde  $\omega_0 = 2\pi/T$  je základní frekvence signálu. Koeficienty  $a_0, b_0, c_0$  a  $X_0$  odpovídají stejnosměrné složce signálu.

## 2.5 Fourierova transformace obecného signálu

Rozkladem periodického signálu pomocí FŘ získáme nekonečný počet koeficientů. Ale i s nekonečným počtem koeficientů je výsledné spektrum nespojité. Obsahuje pouze složky s frekvencemi, které jsou násobkem základní harmonické frekvence. Zobecnění Fourierových řad na obecný (tedy i aperiodický) signál je Fourierova transformace dále jen FT.

Principem FT je rozšíření intervalu periodicity  $T$  na interval  $(-\infty; \infty)$ . Toto roztažení intervalu do nekonečna zmenší interval mezi harmonickými složkami až na nulu. Proto je spektrum FT spojitou funkcí frekvence. U periodických signálů tvoří spojitě spektrum FT obálku spektra FŘ.

$$X(\omega) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t) \exp(-j\omega t) dt \quad (2.5)$$

$$x(t) = F^{-1}\{X(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \exp(j\omega t) d\omega \quad (2.6)$$

Funkce úhlové frekvence  $X(\omega)$  je obrazem signálu  $x(t)$  ve frekvenční oblasti. Někdy se také nazývá Fourierovo nebo komplexní spektrum.

### 2.5.1 Fourierova transformace diskrétního signálu (DTFT)

DTFT je duální operací k FŘ a přiřazuje vzorkovanému signálu spojitě periodické spektrum. Je definována následujícím vztahem.

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] \exp(-j\omega n) \quad (2.7)$$

### 2.5.2 Diskrétní Fourierova transformace

DFT je podle (Hlaváč, 2007) aplikace diskrétních Fourierových řad na konečný vzorkovaný signál. Jejím výsledkem je konečný počet frekvenčních vzorků. DFT v sobě ukrývá podmínku periodicity vstupního signálu tím, že analyzovaný signál uvažuje jako by šlo o jednu periodu periodického signálu. A výsledné spektrální vzorky jsou jednou periodou periodického spektra.

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi nk/N) \quad (2.8)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp(-j2\pi nk/N) \quad (2.9)$$

V některé literatuře se uvádějí  $X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \exp(-j2\pi nk/N)$  (2.10)  $x[n] = \sum_{k=0}^{N-1} X[k] \exp(-j2\pi nk/N)$  (2.11), kde je přesunut koeficient  $1/N$ . Takto definovanou DFT používají například analyzátory firmy Bruel & Kjaer.

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \exp(-j2\pi nk/N) \quad (2.10)$$

$$x[n] = \sum_{k=0}^{N-1} X[k] \exp(-j2\pi nk/N) \quad (2.11)$$

DFT lze také zapsat maticově

$$[X_0, X_1, \dots, X_{N-1}] = [x_0, x_1, \dots, x_{N-1}] \cdot \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \dots & W_N^{(N-1)} \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)^2} \end{bmatrix} \quad (2.12)$$

$$\text{Kde je } W_N^{nk} = \exp\left(-\frac{j2\pi nk}{N}\right). \quad (2.13)$$

Hodnoty  $X[k]$  představují komplexní spektrum na frekvencích:

$0, F_s/N, 2F_s/N, \dots, (N-1) F_s/N$ . Pro reálný vzorkovaný signál jsou hodnoty koeficientů  $F_k$  a  $F_{N-k}$  komplexně sdružené.

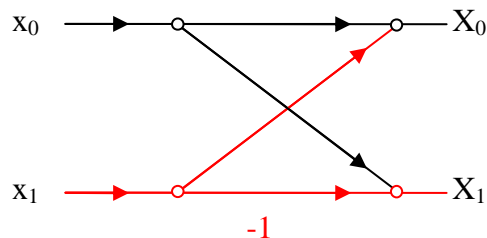
### 2.6 Rychlá Fourierova transformace FFT

V roce 1965 pánové J. W. Cooley a J. W. Tuckey popsali úsporný způsob výpočtu DFT Spočívá v postupné decimaci vstupní posloupnosti délky  $2^N$  až k délce 2,



kde je rovnici  $X[k] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi nk/N)$  (2.8) možné snadno vyčíslit. Výpočet je možné znázornit grafem na obrázku 2.1, který se nazývá motýlek. Data jsou před vstupem do motýlku násobena otáčecími faktory (twiddle factors), které odpovídají členu  $e^{-j2\pi nk/N}$ . Výpočet probíhá ve vstupním poli (angl. In Place). Cooley a Tuckey použili decimaci číslem 2 v časové oblasti. Tato metoda se anglicky nazývá Radix-2 DIT jak napsal (Hlaváč, 2007). V aplikacích se vyskytují i algoritmy s vyšším základem (Radix-4,8,16,...), a dále algoritmy s kombinací různých základů (Split Radix)

Motýlky s vyššími základy mají podle teorie menší výpočetní složitost. Prakticky však vyžadují větší počet „režijních“ operací, čímž se jejich výhodnost poněkud zmenšuje.



Obr. 2.1 Motýlek

## 2.6.1 Odvození FFT

Algoritmus FFT využívá diagonální a zrcadlové symetrie členů  $W_N^{nk}$  v matici ve

$$\text{vztahu } [X_0, X_1, \dots, X_{N-1}] = [x_0, x_1, \dots, x_{N-1}] \cdot \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \dots & W_N^{(N-1)} \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)^2} \end{bmatrix}$$

(2.12). N-bodovou DFT lze vypočítat jako N/2-bodovou DFT zvlášť vypočtených pro sudé a liché vzorky.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} = \quad (2.14)$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[2n] W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] W_N^{(2n+1)k} = \quad (2.15)$$

$$= \sum_{n=0}^{\frac{N}{2}-1} x[2n] W_N^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] W_N^{nk} = \quad (2.16)$$

$$= X_s[k] + W_N^k X_L[k] \quad (2.17)$$

Protože DFT je periodická a zrcadlově symetrická funkce, lze výpočet druhé (horní) poloviny vzorků od  $N/2$  do  $N-1$  získat z hodnot vzorků první (spodní) poloviny od 0 do  $N/2-1$ .

$$X[k + N/2] = X[k] \quad (2.18)$$

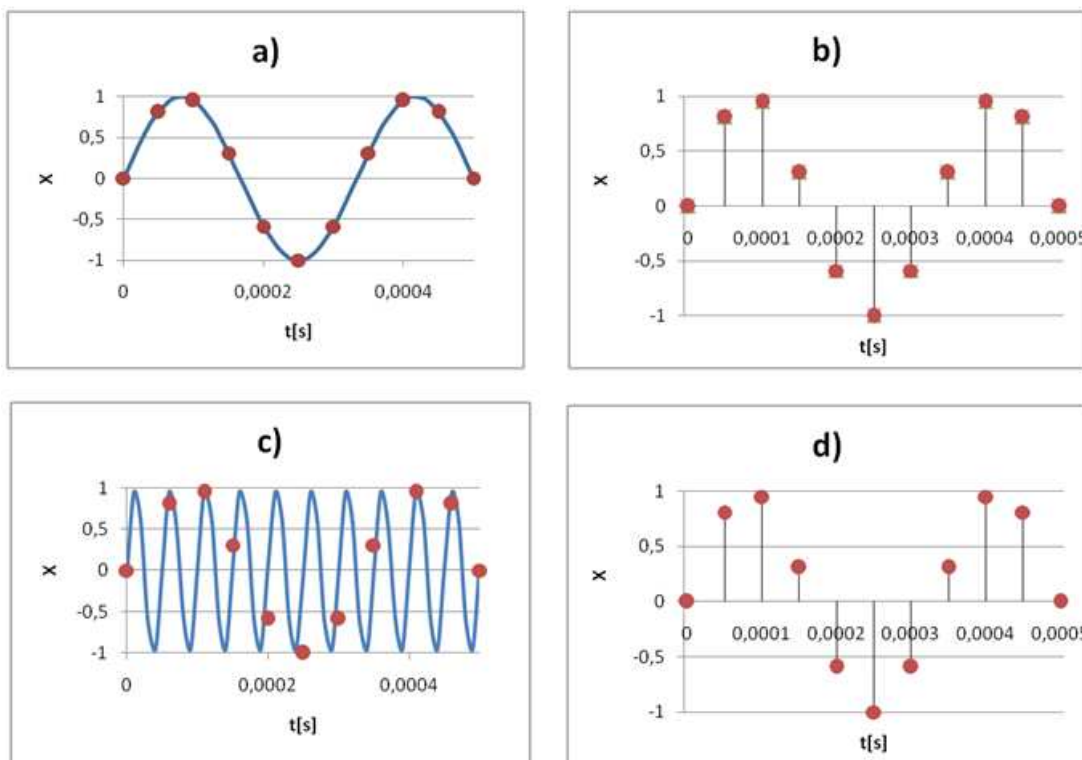
Toto dělení na polovinu lze rekurzivně aplikovat až na dvoubodovou DFT. Která je nejjednodušším případem DFT.

$$X[0] = x[0] + x[1] \text{ a } X[1] = x[0] - x[1] \quad (2.19)$$

$$\text{Protože } \exp\left(-\frac{j4\pi}{2}\right) = 1 \text{ a } \exp\left(-\frac{j2\pi}{2}\right) = -1 \quad (2.20)$$

## 2.7 Aliasing

Pokud není splněn vzorkovací teorém, tedy pokud signál obsahuje složky o frekvencích vyšších než je polovina vzorkovací frekvence, tak dochází k superpozici dvou replik spektrální obálky periodicky se opakujícího spektra. Problém však není v posloupnosti vzorků a její Fourierově transformaci. Fourierova transformace je vzájemně jednoznačná. Posloupnost vzorků, kterou získáme inverzní transformací z Fourierova obrazu, je totožná s původní posloupností vzorků signálů. Nedodržením vzorkovacího teorému ztrácíme informaci v okamžiku vzorkování. A k vzorkům již nelze najít správný časový průběh. Na obr. 2.2 a,b je sinusový signál s frekvencí 3 kHz, navzorkovaný frekvencí 20kHz. Na obrázcích 2.2 c,d je sinusový signál s frekvencí 17 kHz vzorkovaný opět s frekvencí 20 kHz. Jak je z obrázků vidět frekvence 17kHz se přeložila na frekvenci 3 kHz. Vzorky jsou tudíž u obou signálů stejné.



## 2.2 Vznik aliasingu

Z tohoto důvodu jak líše (Randall, 1987) jsou ve všech analyzátoch a měřících zařízeních nainstalovány filtry, typu dolní propust, které nepropustí do vzorkovacího A/D převodníku signály s vyšší frekvencí než je polovina vzorkovací frekvence.

Obecné filtry typu dolní propust filtry mohou být jak analogové, tak i digitální. Filtr před A/D převodníkem je samozřejmě analogový. Parametry těchto filtrů odpovídají frekvenčnímu rozsahu daného zařízení. Pokud má zařízení umožňovat vzorkování signálu s různou vzorkovací frekvencí, musí obsahovat buď více analogových filtrů (pro každou frekvenci jeden) nebo vzorkovat signál maximální vzorkovací frekvencí. Pro snížení vzorkovací frekvence pak použijeme digitální filtraci a následné převzorkování signálu.

Protože při Fourierově transformaci vypočteme hodnoty všech frekvencí až do poloviny vzorkovací frekvence. Takto získané hodnoty nechceme mít ovlivněné frekvenční charakteristikou filtru. Jsou v těchto přístrojích použity velmi příkré filtry s mezní frekvencí okolo 80% Nyquistovy frekvence. Z vypočtených hodnot se pak zobrazují pouze hodnoty neovlivněné filtrem. Typicky při délce záznamu 1024 vzorků se vypočte 512 frekvenčních složek a 400 se jich zobrazí. Počet čar je tedy přibližně roven délce záznamu vydělené číslem 2.56.

Použití dolno-propustních filtrů při analýze stacionárních signálů nám zabrání vzniku aliasingu. Proto je aliasing z uživatelského hlediska ve většině případů nepodstatný. Problém, se kterým je nutno se vypořádat, nastává při použití řádové analýzy<sup>1</sup> (angl. “tracking analysis“), kdy se mění vzorkovací frekvence v závislosti např. na otáčkách stroje.

---

<sup>1</sup> Někdy se také nazývá souběhová filtrace. Je to metoda při které se mění vzorkovací frekvence v závislosti na velkém signálu např. z čidla otáček. Metoda se používá při analýze nestacionárních signálů. Jako jsou rozběhy a doběhy strojů.

### 3 Vyhodnocování signálů

U signálů můžeme vyhodnocovat dva základní druhy charakteristik. Prvním druhem jsou souhrnné charakteristiky, které nám dávají hodnotu popisující signál jako celek. Takovouto hodnotou je například efektivní hodnota signálu při měření hluku a vibrací. Druhým druhem jsou frekvenční spektra. Frekvenční spektrum je funkce frekvence a zjednodušeně řečeno představuje míru zastoupení složek o určité frekvenci v analyzovaném signálu.

Při měření frekvenčních spekter signálů nás často zajímá, jak velkých amplitud dosahuje analyzovaný signál na určitých frekvencích nebo v určitých frekvenčních pásmech. Kmitočtové složení signálu může být při měření získáno pomocí filtrů (pásmových propustí). Tyto filtry nám propustí do vyhodnocovacího zařízení pouze signály určité frekvence. Může se jednat o skupinu filtrů s pevně danými středními kmitočty, nebo jeden přeladitelný filtr, u kterého měníme střední kmitočet. U reálného filtru neexistuje jediná hraniční frekvence mezi propustným a nepropustným pásmem, ale je zde celé přechodové pásmo. Pro další úvahy budeme zjednodušeně předpokládat ideální filtr bez přechodového pásma se spodní mezní frekvencí  $f_s$  a horní  $f_h$ . S jejich pomocí můžeme definovat absolutní šířku propustného pásma jako  $\Delta f = f_h - f_s$  a relativní šířku jako  $\Delta f / f_c$ .

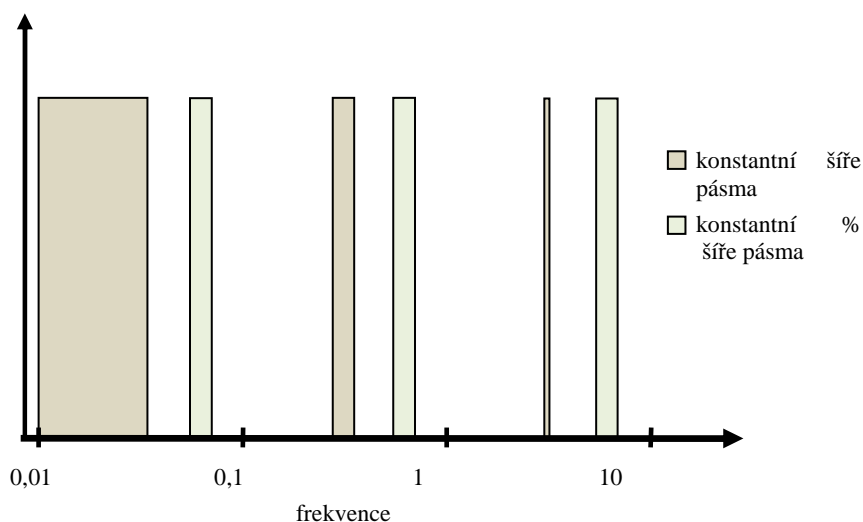
Tyto vztahy jsou základem dvou hlavních typů kmitočtové analýzy. Jedná se o analýzu s procentuálně konstantní šíří pásma nebo analýzu s konstantní šíří pásma. Efekt, který má šíře kmitočtového pásma na výsledné spektrum, je vyjádřen na obrázku obr.3.1 a na obr.3.2. Při analýze signálu s konstantní relativní šířkou pásma tzv. CPB (Constant Percentage Bandwidth), je šíře pásma propustnosti procentuálně konstantní vzhledem ke střednímu kmitočtu v pásmu, takže absolutní hodnota šíře pásma se zvětšuje s rostoucím středním kmitočtem v pásmu. Jako šířka pásma se volí jedna oktáva nebo její zlomky. CPB analýza je charakteristická použitím logaritmické frekvenční osy.

Druhá metoda je metoda s konstantní šíří pásma. U této metody šíře pásma propustnosti nezávisí na středním kmitočtu pásma, ale je v celém měřeném rozsahu stejná. Tato metoda je charakteristická lineární frekvenční osou. Používá např. u FFT analyzátorů.

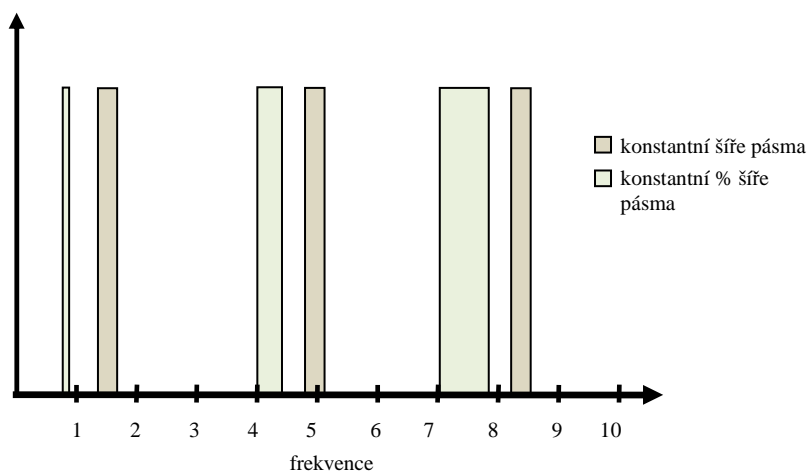
Při diagnostice strojů jsou v analyzovaných signálech obsaženy složky závislé

na činnosti stroje, například hluk nebo vibrace vzniklé otáčení hřídelů a ozubených kol. Pro další vývoj těchto strojů a odstranění jejich vad chceme u těchto složek znát jejich přesnou frekvenci a amplitudu. Proto se takovéto signály zkoumají analýzou s konstantní šíří pásma.

Na druhé straně měření hluku, která mají sloužit k určení celkové hlučnosti stroje, obvykle nevyžadují přesnou znalost spektra včetně úrovně diskrétních složek. Například nás zajímá, jestli signál obsahuje složku 50 nebo 100 Hz a jakou má tato složka amplitudu. A příliš se nestaráme o to, jakou přesnou frekvenci má složka na 1 kHz jestli to je 950, nebo 1000 Hz. V takovýchto případech volíme analýzu s procentuálně konstantní šířkou pásma a logaritmickou frekvenční osou.



*Obr. 3.1 Šířka pásma s logaritmickou frekvenční osou*



*Obr. 3.2 Šířka pásma s lineární frekvenční osou*

## 3.1 Výpočet amplitudového a fázového spektra signálu

### 3.1.1 Výkonové spektrum

Je jedna ze základních charakteristik ve spektrální oblasti. Určuje výkon konkrétní harmonické složky signálu. Výkonové spektrum nám nedává žádnou informaci o fázi signálu. Fázová informace získaná Fourierovou transformací je vztažena k počátku časového záznamu signálu. Sinusová vlna ukazuje fázi  $-90^\circ$  na příslušné frekvenci a kosinusová vlna ukazuje fázi  $0^\circ$ . Při analýze signálů obvykle sledujeme relativní fázi mezi dvěma složkami nebo fázový rozdíl mezi dvěma simultánně získanými signály.

Pomocí Fourierovy transformace získáme dvoustranné spektrum v komplexním tvaru s reálnou a imaginární částí. Převodem spektra na polární tvar získáme magnitudu a fázi. Amplituda jednotlivých složek spektra souvisí s počtem vzorků signálu v časové oblasti. Pro výpočet amplitudy a fáze z dat získaných Fourierovou transformací se používají následující vztahy:

$$\text{Amplituda} = \frac{\text{Magnituda}(X_k)}{N} = \frac{\sqrt{(re(X_k))^2 + (im(X_k))^2}}{N} \quad (3.1)$$

$$\text{Fáze} = \arctg\left(\frac{im(X_k)}{re(X_k)}\right) \quad (3.2)$$

Kde funkce arkus tangens vrací hodnotu fáze v intervalu od  $-\pi$  do  $+\pi$  radiánů. Dvoustranné amplitudové spektrum ve skutečnosti ukazuje polovinu amplitudy v pozitivních frekvencích a druhou polovinu amplitudy v negativních frekvencích. Pro převod spektra na jednostranné je potřeba každou složku z první poloviny spektra (mimo stejnosměrné) vynásobit dvěma. Druhá polovina spektra nám neposkytuje žádné další informace, a proto není potřeba.

### 3.1.2 Efektivní hodnota amplitudy a fáze ve stupních

Pro převod amplitudového spektra na efektivní hodnoty je potřeba vydělit všechny složky spektra (mimo stejnosměrné) odmocninou ze dvou, protože již při převodu na jednostranné spektrum jsou tyto složky jednou násobeny dvěma. Je možné tyto dvě operace spojit a vypočítat jednostranné spektrum středních hodnot přímo z první poloviny spektra reálného signálu násobením odmocninou ze dvou.

$$V_{rms} = \sqrt{2} \frac{Magnituda(X_k)}{N} \text{ Pro } k = 1 \text{ až } \frac{N}{2} - 1 \quad (3.3)$$

$$V_{rms} = \frac{Magnituda(X_k)}{N} \text{ Pro } k = 0 \quad (3.4)$$

Kde  $k$  je číslo frekvenční čáry nebo index v poli hodnot z Fourierovy transformace.

Převod fázového spektra v radiánech na stupně je dán vztahem:

$$\varphi(k)[^\circ] = \frac{180}{\pi} \cdot \varphi(k) \quad (3.5)$$

### 3.1.3 Převod do logaritmických jednotek

Některé přístroje nebo programy mohou zobrazovat spektrum signálu jak v lineárním měřítku, tak i v decibelech. Lineární měřítko zobrazuje amplitudy tak, jak odpovídají skutečnosti. Při zobrazení spektra v decibelech jsou hodnoty převedeny na logaritmické měřítko. Ve vztahu  $dB = 10 \log_{10}(\frac{P}{P_r})$  (3.6) jsou decibely definovány pro výkonové spektrum a ve vztahu  $dB = 20 \log_{10}(\frac{A}{A_r})$  (3.7) jsou definovány pro amplitudové spektrum.

$$dB = 10 \log_{10}(\frac{P}{P_r}) \quad (3.6)$$

Kde  $P$  je měřený výkon a  $P_r$  je referenční výkon.

$$dB = 20 \log_{10}(\frac{A}{A_r}) \quad (3.7)$$

Kde  $A$  je měřená amplituda a  $A_r$  je referenční amplituda.

$dB = 10 \log_{10}(PP_r)$  (3.6) a  $dB = 20 \log_{10}(AA_r)$  (3.7) pro výpočet amplitudy a výkonu v decibelech vyžadují referenční hodnotu, která slouží jako hladina 0 decibel. K určení referenční hodnoty pro výpočet decibel je možné použít několik běžných konvencí.

- Referenční hodnota jeden volt ( $1 V_{rms}$ ) pro amplitudu respektive ( $1 V_{rms}^2$ ) pro výkon. Výsledné jednotky potom jsou dBV.
- Referenční hodnota 1 mW pro  $50 \Omega$  zátěž pro radiové frekvence, kde 0 dB je  $0.22 V_{rms}$ . Výsledné jednotky potom jsou dBm.



- Referenční hodnota 1 mW pro 600  $\Omega$  zátěž pro radiové frekvence, kde 0 dB je 0.78 V<sub>rms</sub>. Výsledné jednotky potom jsou dBm.
- Referenční hodnota pro hladinu akustického tlaku je 20  $\mu$ Pa.

Při použití amplitudy nebo výkonu jako druhé mocniny amplitudy je výsledná hladina decibel stejná. Násobení decibelů dvěma je stejné jako umocnění amplitudy na druhou. Proto dostaneme stejnou hladinu decibel, ať už použijeme amplitudové nebo výkonové spektrum.

### 3.1.4 Zobrazení výsledků v decibelovém měřítku

Zobrazení amplitudového nebo výkonového spektra v decibelovém měřítku nám umožní vidět široký rozsah funkčních hodnot. Můžeme vidět malé složky signálu v přítomnosti velkých složek. Například předpokládejme, že chceme zobrazit signál, který obsahuje amplitudy od 0,1V do 100V na zařízení, které má displej o výšce 10cm. Při použití lineárního měřítka zařízení zobrazí 100 V amplitudu na celou výšku displeje. To znamená, že zařízení zobrazuje 10V na jeden centimetr výšky displeje. Amplituda 0,1 V je pak zobrazena pouze na 0,1mm. Protože čára o výšce 0,1 mm je stěží viditelná, tak je velice snadné takovou složku signálu přehlédnout. Při zobrazení signálu v logaritmickém měřítku bude mít 100 V amplituda 40dB a amplituda 0,1 V bude -20 dB.

Následující tabulka 2.1 ilustruje vztah mezi amplitudou výkonem a hladinou dB. Tabulka ukazuje, jak je možné vměstnat velký rozsah amplitud do malé množiny čísel použitím logaritmického měřítka.

*Tab. 3.1 Rozsah amplitud v logaritmickém měřítku*

dB	Výkon	Amplituda
+40	10,000	100
+20	100	10
+6	4	2
+3	2	1.4
0	1	1
-3	1/2	1/1.4
-6	1/4	1/2
-20	1/100	1/10
-40	1/10,000	1/100

### 3.1.5 Odhad výkonu a frekvence

Pokud frekvenční složka leží mezi dvěma frekvenčními čarami, pak se energie této složky ve spektru rozlije mezi sousední čáry a její amplituda klesne. Skutečná špička pak leží někde mezi dvěma frekvenčními čarami. Skutečnou frekvenci složky můžeme odhadnout z hodnot sousedních složek. Pomocí váženého průměru složek okolo zkoumaného vrcholu. Získáme tak větší frekvenční rozlišení než je  $\Delta f$ , které nám dává Fourierova transformace. Výpočet je dán vztahem:

$$\text{Odhadovaná frekvence} = \frac{\sum_{i=j-3}^{j+3} (P(i)(i\Delta f))}{\sum_{i=j-3}^{j+3} (P(i))} \quad (3.8)$$

Kde je index vrcholu, který zkoumáme.

Rozpětí  $j \pm 3$  je použito, protože představuje rozsah širší než je šíře hlavních laloků vyhlazovacích oken (obdélník, Hanning, Hamming, Blackman-Harris, Exact Blackman, Blackman a Flat Top).

Odhad výkonu zkoumané složky můžeme provést sečtením jednotlivých složek okolo, jinými slovy vypočteme plochu pod tímto vrcholem. Jak je uvádí (NI The Fundamentals of FFT). Pro odhad výkonu můžeme použít následující vztah:

$$\text{Odhadovaný výkon} = \frac{\sum_{i=j-3}^{j+3} P(i)}{enbw} \quad (3.9)$$

## 3.2 Oktávové kmitočtové pásmo

Kmitočtové pásmo o šířce jedné oktávy je charakterizováno poměrem krajních frekvencí omezujících oktávu. Jak je definováno v (Nový, 2000).

$$\frac{f_h}{f_s} = 2 \quad (3.10)$$

Každou oktávu označujeme střední frekvencí  $f_c$ , tuto frekvenci můžeme určit ze vztahu  $f_c = \sqrt{f_s f_h}$  (3.11).

$$f_c = \sqrt{f_s f_h} \quad (3.11)$$

Střední kmitočty v oktávových pásmech jsou

Spodní frekvence pásma je dána vztahem  $f_s = \frac{f_c}{\sqrt{2}}$  (3.12).

$$f_s = \frac{f_c}{\sqrt{2}} \quad (3.12)$$

Zatímco pro horní frekvenci oktávového pásma použijeme výraz

$$f_h = f_c \sqrt{2} \quad (3.13)$$

Tab. 3.2 Střední oktávové a 1/3 oktávové kmitočty pro měření v akustice

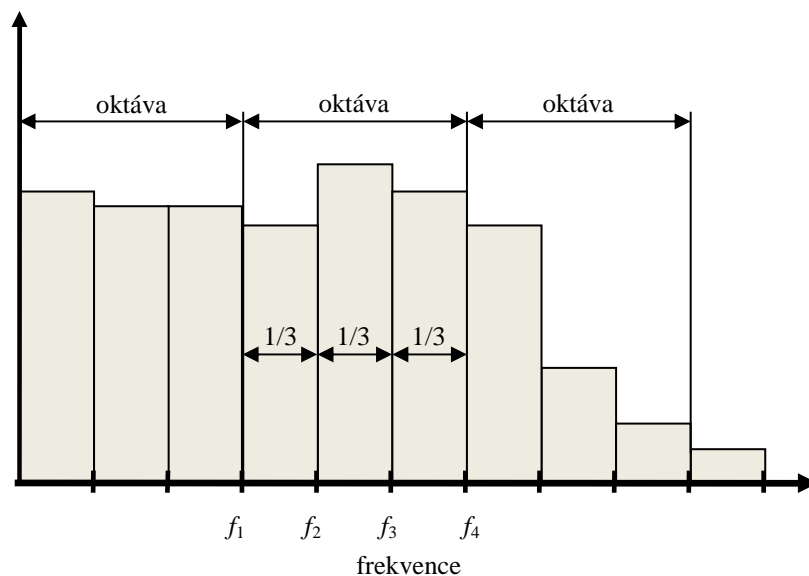
Střední kmitočet [Hz]	Oktáva	1/3 oktávy	Střední kmitočet [Hz]	Oktáva	1/3 oktávy	Střední kmitočet [Hz]	Oktáva	1/3 oktávy
25		X	250	X	X	2500		X
31,5	X	X	315		X	3150		X
40		X	400		X	4000	X	X
50		X	500	X	X	5000		X
63	X	X	630		X	6300		X
80		X	800		X	8000	X	X
100		X	1000	X	X	10000		X
125	X	X	1250		X	12500		X
160		X	1600		X	16000	X	X
200		X	2000	X	X	20000		X

$$(f_h - f_s) = f_c \left( \sqrt{2} - \frac{1}{\sqrt{2}} \right) = \frac{f_c}{\sqrt{2}} \quad (3.14)$$

Porovnání šíře oktávového pásma vzhledem k pásmům konstantní šíře je provedeno na obrázku 3.1. Kdybychom oktávová pásma očíslovali vzestupně, tak že 1. oktávou by byla oktáva  $f_c=31,5$  Hz a poslední 10. oktáva by byla na kmitočtu  $f_c=16$  kHz, platí pro výpočet středních kmitočtů vzorec  $f_c = 15,625 \cdot 2^n$  (3.15).

$$f_c = 15,625 \cdot 2^n \quad (3.15)$$

Kde n je číslo oktávy.



Obr. 3.3 Oktávové a třetinooktávové spektrum

### 3.3 Třetinooktávové pásmo

Rozdělíme-li oktávové pásmo na třetiny (v logaritmických stupnicích), tím získáme třetinooktávové pásmo viz. obrázek 3.3. Pro každé třetinooktávové pásmo musí platit následující závislost. Je-li frekvencemi  $f_1$  a  $f_4$  ohraničeno pásmo jedné oktávy a frekvencemi  $f_2$  a  $f_3$  krajní frekvence vnitřní třetiny oktávy, můžeme psát:

$$\log \frac{f_2}{f_1} + \log \frac{f_3}{f_2} + \log \frac{f_4}{f_3} = \log \frac{f_4}{f_1} = \log 2 \quad (3.16)$$

Kde platí rovnost

$$\frac{f_2}{f_1} = \frac{f_3}{f_2} = \frac{f_4}{f_3} \quad (3.17)$$

Poměr krajních kmitočtů v libovolné třetině oktávy je konstantní

$$\frac{f_2}{f_1} = \sqrt[3]{2} = 1,26 \quad (3.18)$$

Zásadu, která platí pro poměr krajních frekvencí v oktávě nebo v 1/3 oktávy, musíme uplatnit i při výpočtu středních kmitočtů v určitém pásmu.

### 3.4 Způsoby výpočtu CPB analýzy

Pro získání spektra s konstantní procentuální šířkou propustnosti neexistuje žádná přímá matematická definice. CPB analýza totiž není matematická transformace, jako je například Fourierova, nebo Laplaceova transformace. CPB analýza je většinou popisována pomocí skupiny filtrů, tedy tak jak v minulosti vznikla. Dnes se pro syntézu CPB analýzy používají dva způsoby. Prvním způsob pro získání CPB spektra používá skupinu pásmových propustí. Tento způsob používá ve svých analyzátoch firma *Brüel & Kjær*. Odlišnou cestou se vydala konkurenční firma National Instrument, jejíž způsob výpočtu spektra vychází z Fourierovy transformace.

#### 3.4.1 CPB syntézou z FFT

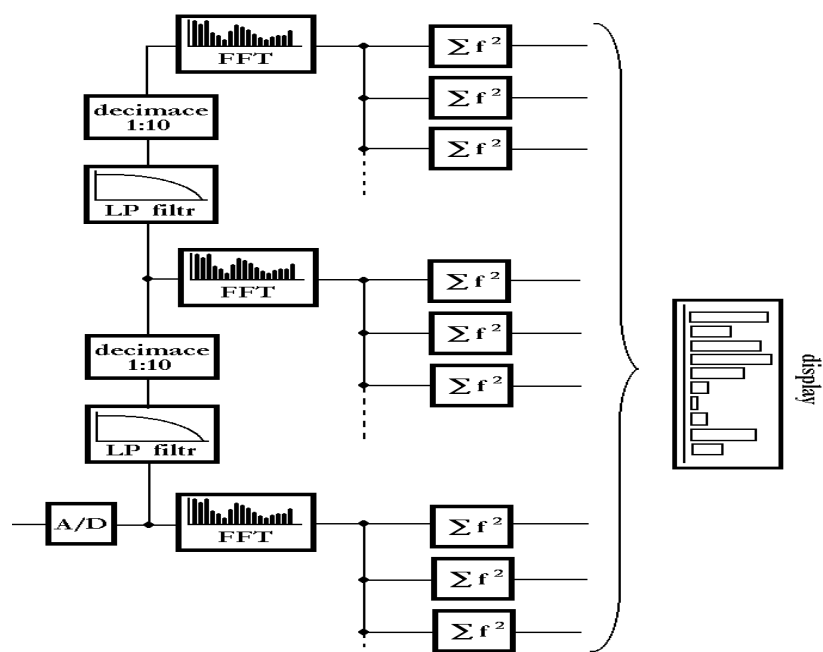
Nejjednodušší způsob výpočtu oktávových a třetinooktávových spekter *CPB* se uskutečňuje prostřednictvím výpočtu FFT, přičemž výkon v jednotlivých pásmech je získán sumací výkonu složek FFT spektra. Tedy prostým součtem výkonů složek ležících v příslušném pásmu. Tato metoda nám poskytuje jednoduchý a rychlý způsob výpočtu CPB spektra. Avšak má jedno zásadní omezení. Tímto omezením je její frekvenční rozsah. Pro ilustraci tohoto omezení se podívejme na následující příklad.

Budeme-li chtít analyzovat audio signál navzorkovaný běžnou vzorkovací frekvencí 48 kHz je jeho horní pásmo určeno polovinou vzorkovací frekvence. Tedy přesněji je dáno frekvencí vypočtenou podle následujícího vztahu  $f_s/2,56 = 48000/2,56 = 18750$ , k této frekvenci vybereme z tab. 3.1. odpovídající pásmo. Tím je pásmo se středním kmitočtem 16 kHz. Při použití běžné délky záznamu 2048 vzorků vypočteme FFT s frekvenčního rozlišením  $\Delta f = f_s/N = 23,4$ . Opět se podíváme do tab. 3.1 a z rozdílů středních frekvencí najdeme odpovídající pásmo. Tím je pásmo o střední frekvenci 100 Hz. Při analýze tohoto signálu jsme tedy schopni vypočítat 22 třetinooktávových pásem. Rozsah tohoto spektra je tedy i s krajními frekvencemi od 89 Hz do 17800 Hz. Další nevýhodou této metody je diametrálně odlišný počet složek pro výpočet jednotlivých pásem. V našem případě jsou první dvě pásma dány hodnotou pouze jediné frekvenční čáry, a naopak poslední pásmo je dáno součtem více jak 150 hodnot. Pokud snížíme vzorkovací frekvenci na  $f_s = 32 \text{ kHz}$  a zdvojnásobíme délku záznamu, posuneme spodní pásmo hodnotu 31.5 Hz. Snížení vzorkovací frekvence a zvětšení délky záznamu nám posunulo hranici prvního pásma směrem k nižším frekvencím. Při zvětšování délky záznamu brzy narazíme na

výkonnostní strop. Snížení vzorkovací frekvence nám posune hranici jak spodního tak i horního pásma.

### 3.4.2 CPB vypočtené kaskádní FFT

Princip metody podle (NI Third Octave Analysis Toolkit, 1995) vychází ze složení více FFT jak je naznačeno ve výše uvedeném příkladu. Výsledné spektrum je složeno z více FFT, kde každá FFT pracuje se signálem nevzorkovaným jinou vzorkovací frekvencí. Většina hardwarových zařízení má ovšem omezené možnosti vzorování tak, že v daném okamžiku mohou vzorkovat pouze jednou frekvencí. Vzorkovací frekvence se tedy volí podle nejvyšší frekvence, kterou chceme analyzovat. Ostatní se získají úpravou (decimací) již navzorkovaných dat, tento postup se nazývá kaskádní (vícestupňová FFT). A je uveden na obrázku 3.4.



Obr. 3.4 Blokové schéma výpočtu CPB pomocí kaskádní FFT

Celý postup funguje tak, že z navzorkovaných dat vypočteme první skupinu pásem s nejvyššími frekvencemi. Stejná data jdou zároveň do číslicového filtru, který má desetinou mezní frekvenci oproti prvnímu filtru. Z těchto dat je následně vybrán každý desátý vzorek, ostatní vzorky nejsou použity. Po načtení potřebné délky signálu je z dat vypočtena FFT. Z těchto hodnot jsou sumací vypočtena prostřední pásma. Stejný postup se opakuje i ve třetím stupni.

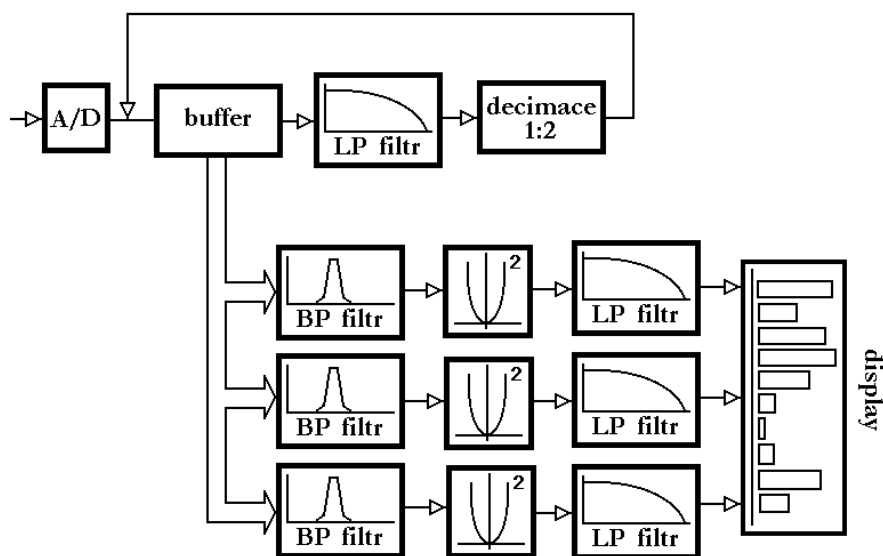
Decimace signálu snižuje jeho frekvenci, ale i zkracuje jeho délku. My však musíme zachovat stejnou délku záznamu pro FFT ve všech stupních. Proto každý další

stupeň potřebuje desetkrát více vzorků než stupeň předcházející. Z každé FFT získáme  $N/2$  hodnot jdoucích od nuly do Nyquistovy frekvence. Z těchto hodnot nevyužijeme přibližně 1/10 frekvencí na nejnižších pořadnicích, protože je s desetkrát lepším rozlišením produkuje následující stupeň.

### 3.4.3 CPB vypočtené pomocí skupiny filtrů

Tato metoda vychází z historického modelu CPB, kdy se spektrum získávalo skupinou analogových filtrů, nebo pomocí jednoho přeladitelného filtru. V dnešní době jsou analogové filtry nahrazeny digitálními, ale základní princip zůstává stejný.

Silným nástrojem při vytváření banky filtrů se stává decimace. Návrh číslicového filtru je totiž invariantní vůči relativní frekvenci (frekvenci normované vzhledem k frekvenci vzorkovací). Charakteristika číslicového filtru se transformuje poměrem vzorkovacích frekvencí, tedy poměrem  $1/D$  kde  $D$  je řád decimace. Tímto se dá jediný návrh filtru využít pro různé vzorkovací frekvence. V praxi se nejčastěji vyskytuje dvojnásobná decimace. Decimujeme tedy signál po oktávách. Tohoto principu využívá blokové schéma na obrázku 3.5.



Obr. 3.5 Banka filtrů pro výpočet CPB

Vstupní signál, který již prošel antialiasing filtrem je nevzorován a uložen do bufferu, z tohoto bufferu se berou data pro samotný výpočet. Za bufferem následuje dolno-propustní filtr a decimační blok, který ze vstupního signálu vynechává každý druhý vzorek, tím je snížena vzorkovací frekvence na polovinu. Výpočet vlastních pásem je proveden pomocí skupiny filtrů. Jejich počet záleží na tom, na kolik pásem

chceme oktávu rozdělit. Na obrázku 3.5 jsou tři filtry, schéma tedy slouží pro výpočet třetinooktávového spektra. Kvadratickým blokem počítáme výkon v jednotlivých třetinooktávových pásmech a dolno-propustní filtr slouží pro výpočet průměrné hodnoty.

### **3.4.4 Metoda použitá v programu**

Jelikož jedním z hlavních požadavků na funkci programu bylo zpětné skládání signálu ze složek odpovídajících hodnotám jednotlivých CPB pásem. Pro toto skládání jsou potřeba informace nejen o frekvenci a amplitudě složek, ale potřebujeme znát i jejich fázi. Z tohoto důvodu nebylo možné použít klasickou syntézu CPB pomocí filtrů, ze které nelze získat informaci o fázi složek. Tento důvod zúžil výběr na volbu mezi jednoduchou nebo kaskádní FFT.

Nakonec byla zvolena metoda výpočtu pomocí jednoduché FFT. Pro výběr této metody hovořila její větší rychlost a potřeba kratší délky signálu. Vzhledem k použití programu pro výukové účely nebyla rozhodující velká přesnost ani velký rozsah pásem, s větším počtem pásem by byl program méně přehledný. Fáze jednotlivých složek pak byla vypočítána jako vážený průměr jednotlivých složek v daném pásmu, složka s největší amplitudou nejvíce ovlivňuje výslednou fázi.

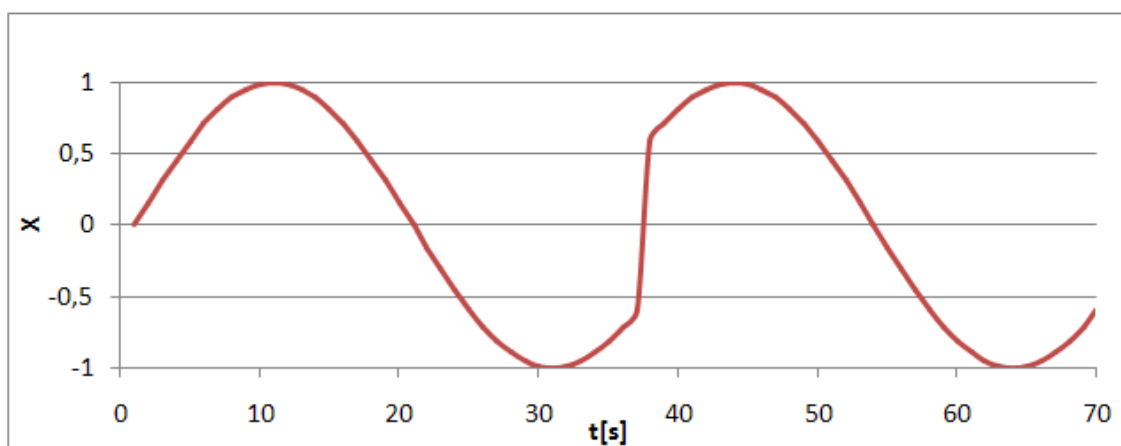


## 4 Rozptyl spektra

Podle Shanonova vzorkovacího teorému lze kompletně rekonstruovat spojitý časový signál z diskrétních ekvidistantně vzorkovaných hodnot. Tedy pokud je nejvyšší frekvence obsažená v časovém signálu nižší než je polovina vzorkovací frekvence (Nyquistova frekvence).

Při reálném vzorkování signálu s konečnou délkou záznamu i při splnění Nyquistova vzorkovacího kritéria se může objevit spektrum, které zdánlivě neodpovídá vzorkovanému signálu. Objeví se rozptýlené spektrum, tento rozptyl spektra způsobuje konečná délka záznamu, jak je uvádí např. (NI Spectral Leakage). Při rozptýleném spektru se zdá, že energie jedné frekvence se rozptýlí do dalších sousedních frekvencí.

Rozptyl spektra je způsoben tím, že algoritmus FFT a DFT předpokládá periodický signál v celém čase s periodou rovnou délce záznamu  $T$ . Při použití algoritmu FFT nebo DFT k výpočtu frekvenčního spektra signálu je použit předpoklad, že záznam signálu je přesně jednou periodou periodického signálu. Konečnost časového záznamu vzorkovaného signálu může způsobit změnu spektrální charakteristiky oproti spektru originálního časově spojitého signálu. Konečnost záznamu může zavést ostré skokové změny do měřených dat. Tyto skokové změny jsou nespojitosti vzniklé napojením neperiodických signálů.



Obr. 4.1 Vznik nespojitostí periodickým prodloužením signálu

Nespojitosti zobrazené na obrázku 4.1 způsobují rozptyl spektra. Výsledné spektrum vzorkovaného signálu je po tom na rozdíl od spektra spojitého signálu roztažené.

Rozptyl spektra se dá vysvětlit i jiným způsobem. Analyzovaný signál o konečné délce si můžeme představit jako nekonečný signál, který je oříznut obdélníkovým oknem. Tedy měříme-li signál  $x(t)$  podobu  $T$ , je tento signál v časové oblasti vždy násoben (hradlován) jednotkovým obdélníkovým oknem  $x_T(t) = x(t)g(t)$ , kde  $g(t)=1$  pro  $0 \leq t \leq T$ , jinak je  $g(t)=0$ . To lze ve frekvenční oblasti zapsat jako  $X_T(f) = X(f) * G(f)$  spektrum je přeměněno konvolucí se spektrem jednotkového okénka  $G(f)$ . Tento jev je neodstranitelný, vzhledem ke konečné délce záznamu obdélníkové okénko existuje vždy.

Výsledné spektrum je pak dáno vztahem:

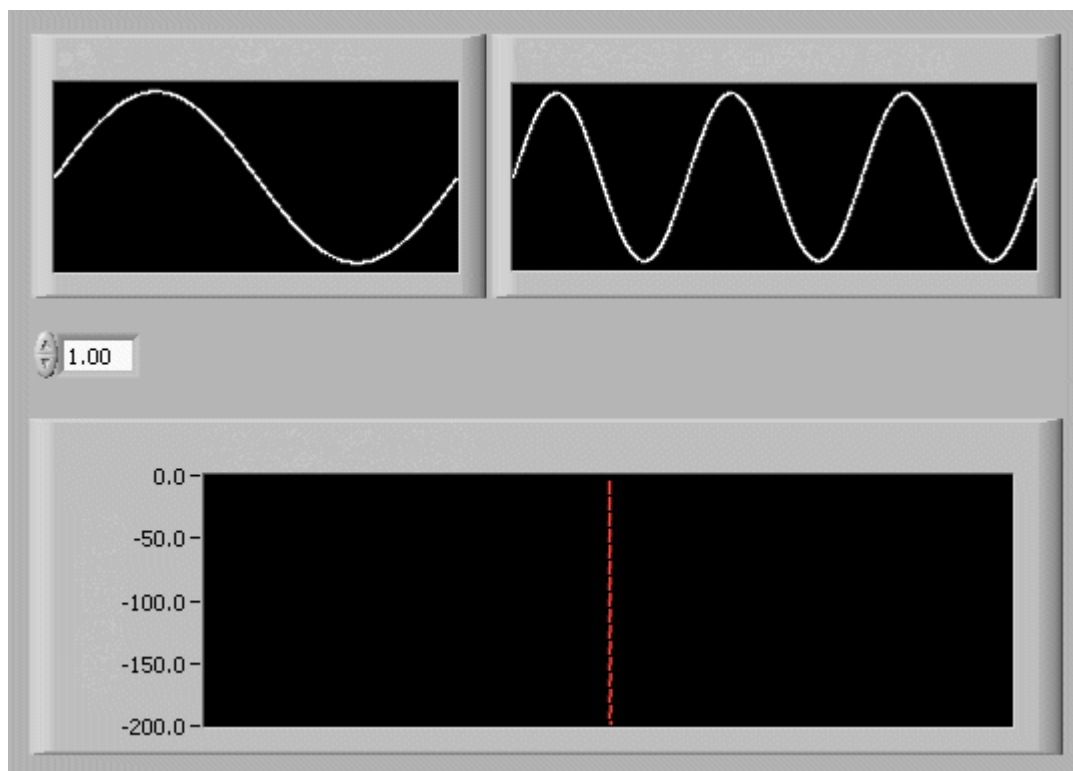
$$X_T(f) = G(f) * X(f) = \int_{-\infty}^{\infty} G(f) \cdot X(f - \varphi) d\varphi \quad (4.1)$$

$$G(f) = \int_{-\infty}^{\infty} g(t) \cdot e^{-j2\pi ft} dt = \int_{-T/2}^{T/2} (\cos(2\pi ft) - j \sin(2\pi ft)) dt \quad (4.2)$$

$$G(f) = T \frac{\sin(\pi f T)}{\pi f T} = T \cdot \text{sinc}(\pi f T) \quad (4.3)$$

### 4.1.1 Signál s celočíselným počtem period

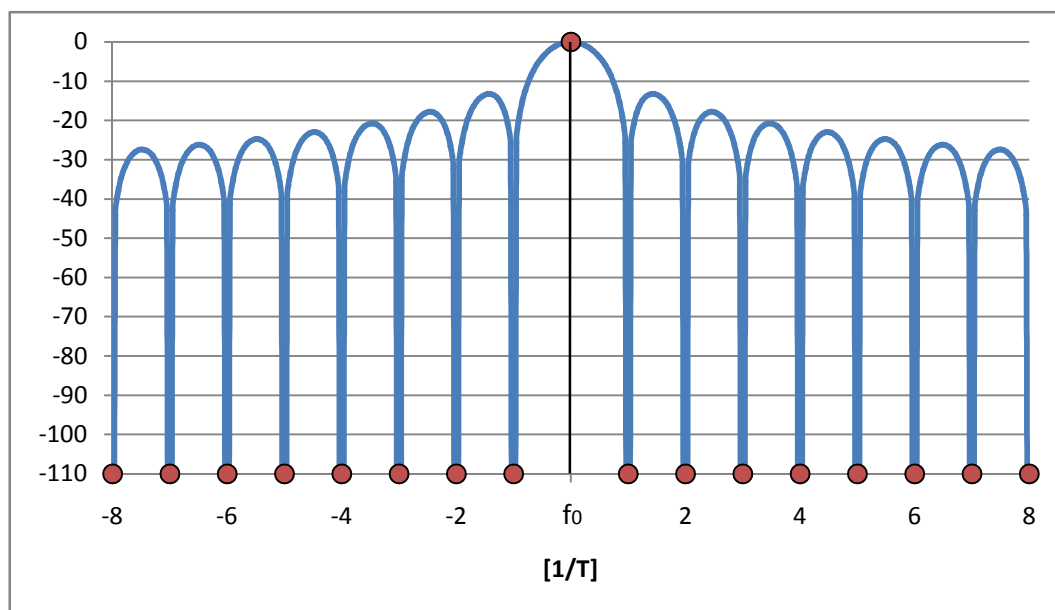
Rozptyl spektra se projeví pouze tehdy, je-li záznam signálu tvořen neceločíselným počtem period signálu. Následující obrázek obr. 4.2 ukazuje záznam signálu s celým počtem period, jeho myšlené periodické prodloužení a Fourierovu transformaci tohoto signálu.



*Obr. 4.2 Signál s celým počtem period*

Na předcházejícím obrázku první okno ukazuje navzorkovaný záznam signálu. Druhé okno znázorňuje periodické prodloužení tohoto signálu, tak jak to předpokládá DFT. A v posledním okně je potom zobrazeno frekvenční spektrum tohoto signálu. Protože signál ve druhém okně neobsahuje žádné nespojitosti, je v něm pouze jedna čára, odpovídající frekvenci a amplitudě sinusového signálu. Aby v signálu nebyly nespojitosti, musí záznam obsahovat pouze celý počet period signálu. Toho se dá dosáhnout pouze tím, že přizpůsobíme vzorkování měřenému signálu.

Jestliže frekvence signálu  $f_0$  je přesně celočíselný násobek  $1/T$ , pak po výpočtu DFT je ve spektru pouze jediná nenulová složka (Tůma, 1997). Ostatní izolované hodnoty jsou v nulách Fourierova obrazu časového okna. Tyto hodnoty jsou pro názornost na obr 4.3 znázorněny červenými značkami na o ose x.



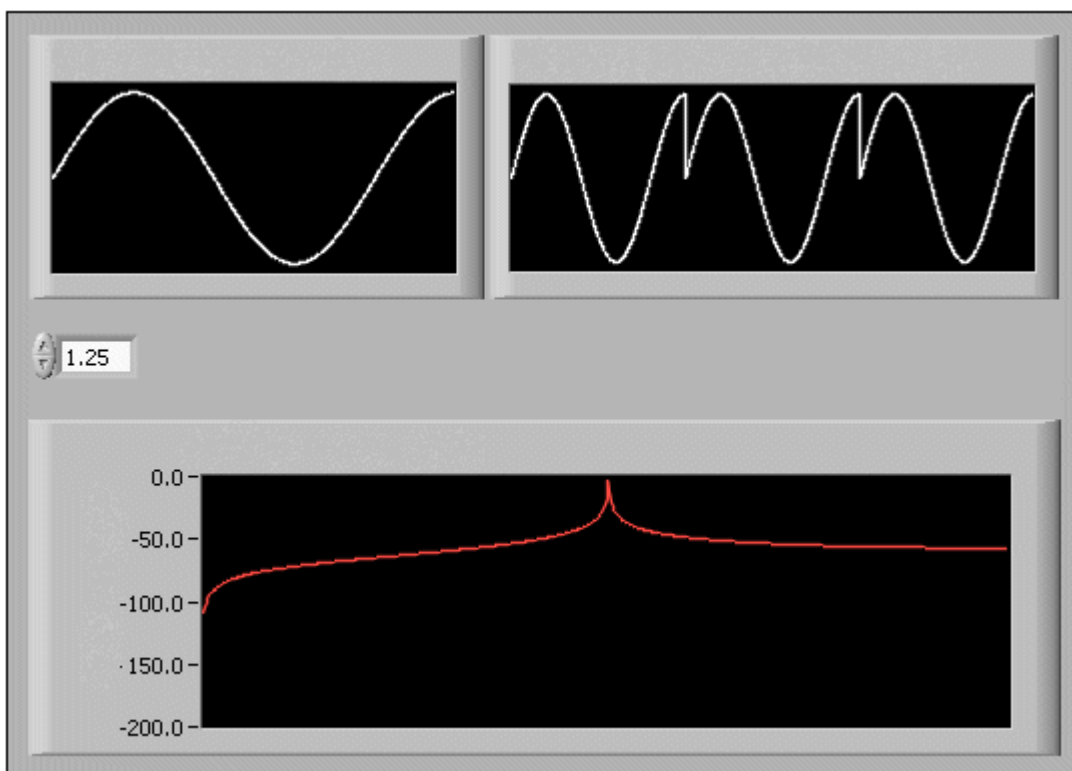
Obr. 4.3 Spektrum signálu o frekvenci  $f_0$ , která je násobkem  $1/T$

#### 4.1.2 Signál s neceločíselným počtem period

I když víme jak analyzovat záznam signálu s celým počtem period, tak velice často nastává situace, kdy potřebujeme analyzovat signál s neúplným počtem period. Většina signálů, které se analyzují v praxi, je stacionární a obsahuje neznámé frekvence. Stacionární signál existuje před, během i po našem měření. Při měření takového signálu nemůžeme zaručit, že nahrajeme signál s úplným počtem period. A pokud signál neobsahuje celý počet period, spektrum signálu je rozptýlené.

Protože Fourierova transformace předpokládá signál periodický v čase. Tak myšleným periodickým prodloužením navzorkovaného signálu vytvoříme umělou nespojitost. Ta se ve spektru může projevit jako velmi vysoká frekvence. Frekvence, která v původním signálu není. Tato vysoká frekvence může být mnohem vyšší než je Nyquistova frekvence a přeloží se na některou frekvenci z intervalu od 0 do  $f_s/2$ . Toto spektrum pak neodpovídá skutečnému spektru původního signálu.

Následující obrázek ukazuje záznam sinusového signálu s neúplným počtem period a Fourierovu transformaci tohoto signálu.

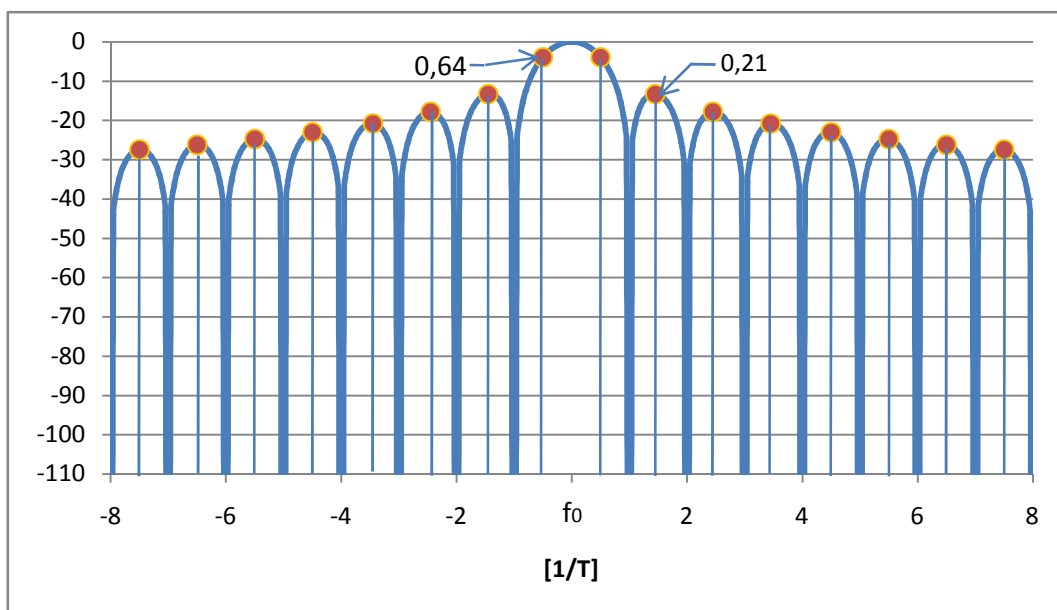


*Obr. 4.4 Signál s neceločíselným počtem period*

Na předchozím obrázku v grafu jedna je zobrazeno 1.25 periody sinusového signálu. V grafu 2 je potom periodické prodloužení tohoto signálu a jsou zde dobře vidět uměle vytvořené nespojitosti. Na grafu 3 je potom vidět spektrum takového signálu. Energie je rozptýlena do širokého rozsahu frekvencí.

Spektrum je rozptýlené, protože pro Fourierovu transformaci používáme konečný vstupní signál s neceločíselným počtem period. K potlačení rozptýlení spektra je možné použít nekonečný záznam od mínus nekonečna do plus nekonečna. S nekonečně dlouhým záznamem FFT vypočte spektrum jako jednu frekvenční čáru o správné frekvenci. Čekání nekonečně dlouhou dobu na nahrání záznamu a na jeho zpracování je nejen nepraktické, ale i nemožné. Proto se pro překonání omezení konečného záznamu používají okénkování funkce, které redukují rozptyl spektra.

Spektrum harmonického signálu, jehož frekvence  $f_0$  není přesně celočíselný násobek  $1/T$ , netvoří jen jediná nenulová složka (Tůma, 1997). Ale po DFT takového signálu vznikne mnoho dalších složek o vysoké amplitudě. Na následujícím obrázku je uveden příklad takového spektra. Frekvence signálu je  $f_0 = (n+0,5)/T$ . Hlavní složky, jejichž frekvence je nejbližší frekvenci  $f_0$ , mají amplitudu rovnu 0,64, místo původní amplitudy rovné jedné. Maximální amplituda postranních složek je rovna hodnotě 0,21.

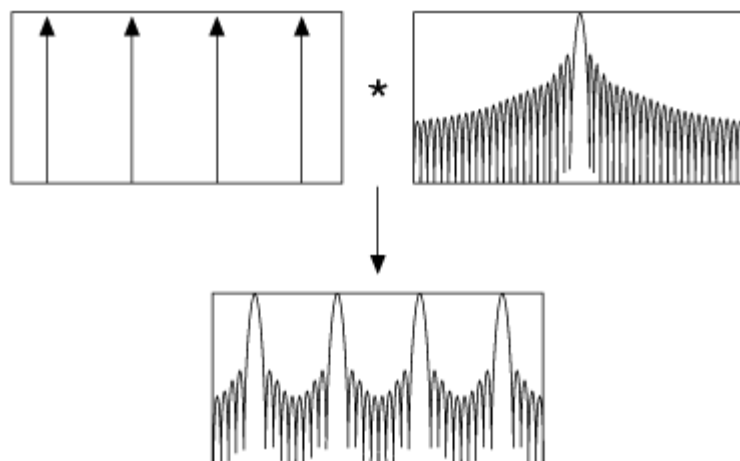


Obr. 4.5 Spektrum signálu o frekvenci  $f_0$ , která leží mezi násobky  $1/T$

## 4.2 Okénkovací funkce

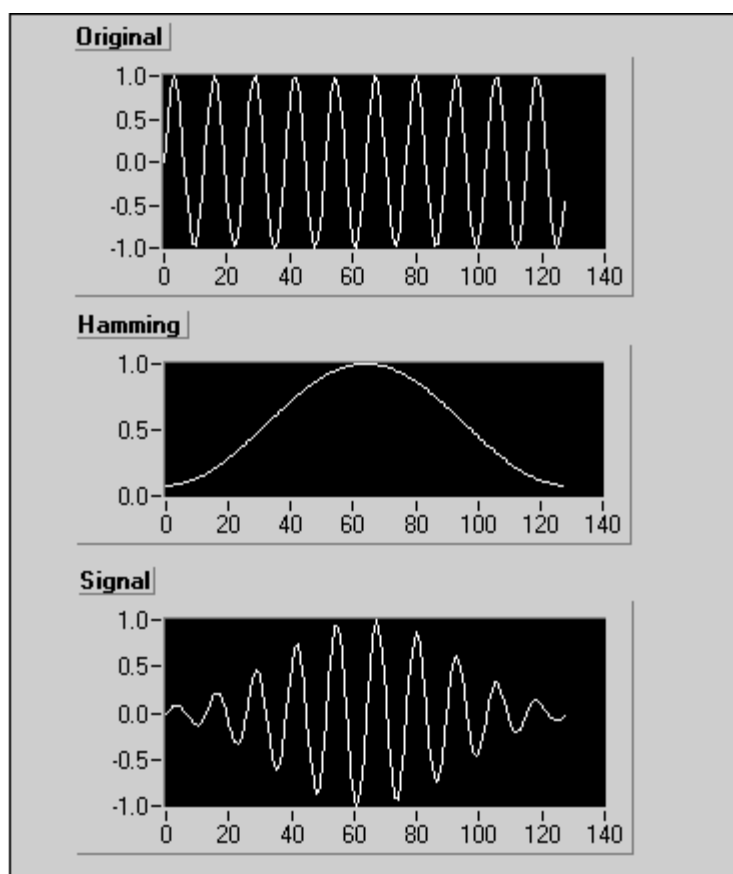
Ke zlepšení spektrálních charakteristik analyzovaného signálu se používají okénkovací funkce, někdy také tzv. vyhlazovací okénka. Jejich funkci si lze představit tak, že minimalizují nespojitosti vzniklé oříznutím části periody signálu a tím zmenšují rozptyl spektra. Míra rozptylu spektra závisí na velikosti nespojitosti. Čím je její amplituda větší, tím je spektrum více rozptýlené a naopak. Vyhlazovací funkce snižují tuto amplitudu na hranicích každého záznamu. To se provádí tak, že časový záznam analyzovaného signálu vynásobíme vyhlazovací funkcí, jejíž amplituda na obou koncích hladce a pozvolně klesá k nule. Délka vyhlazovací funkce se udává v počtu vzorků a je stejná jako délka záznamu zpracovávaného signálu. Místo násobení záznamu signálu v časové oblasti vyhlazovací funkcí je možné provést konvoluci spektra signálu s obrazem okénkovací okénka ve frekvenční oblasti. Tedy spektrum signálu vynásobeného vyhlazovací funkcí je stejné jako spektrum originálního signálu po konvoluci se spektrem vyhlazovací funkce. Tyto funkce mění tvar signálu v časové oblasti, tím ovlivňují i výsledné frekvenční spektrum signálu.

Následující obrázek ukazuje konvoluci spektra originálního signálu se spektrem vyhlazovacího okénka.



*Obr. 4.6 Konvoluce spektra signálu se spektrem okénka*

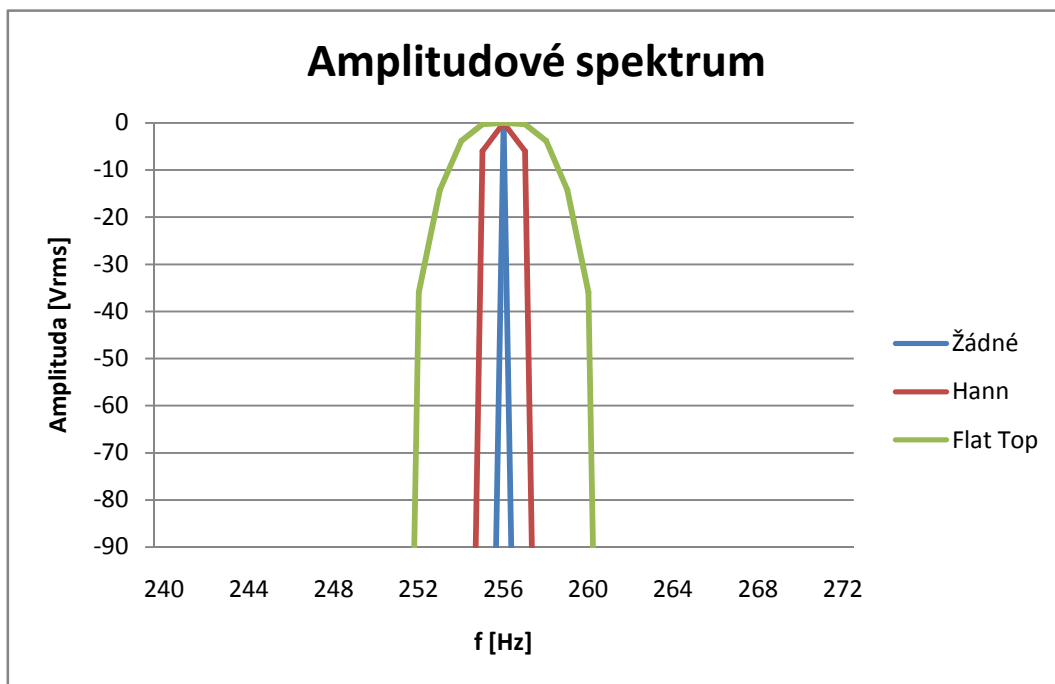
Další obrázek ukazuje vliv Hammingova okénka na časový záznam signálu.



*Obr. 4.7 Vliv Hammingova okénka na časový průběh signálu*

Na předchozím obrázku je časový záznam signálu vážený Hammingovým okénkem. Amplituda signálu se na obou koncích postupně přibližuje k nule. Tím Hammingovo okénko minimalizuje nespojitosti na koncích záznamu. Použitím vyhlazovacího okénka na časový záznam signálu minimalizuje rozptyl spektra.

Následující graf ukazuje efekt použití různých vyhlazovacích okének na spektrum signálu, jehož perioda je celočíselným násobkem  $1/T$ .



Obr. 4.8 Spektrum signálu s celočíselným počtem period

Tab. 4.1 Hodnoty amplitud pro vybrané frekvence

Časové okno	254 [Hz]	255 [Hz]	256 [Hz]	257 [Hz]	258 [Hz]	Max. chyba amplitudy
Obdélníkové (žádné)	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000
Hanning (Hann)	0,0000	0,5000	1,0000	0,5000	0,0000	0,0000
Flat Top	0,6430	0,9665	1,0000	0,9665	0,6430	0,0000

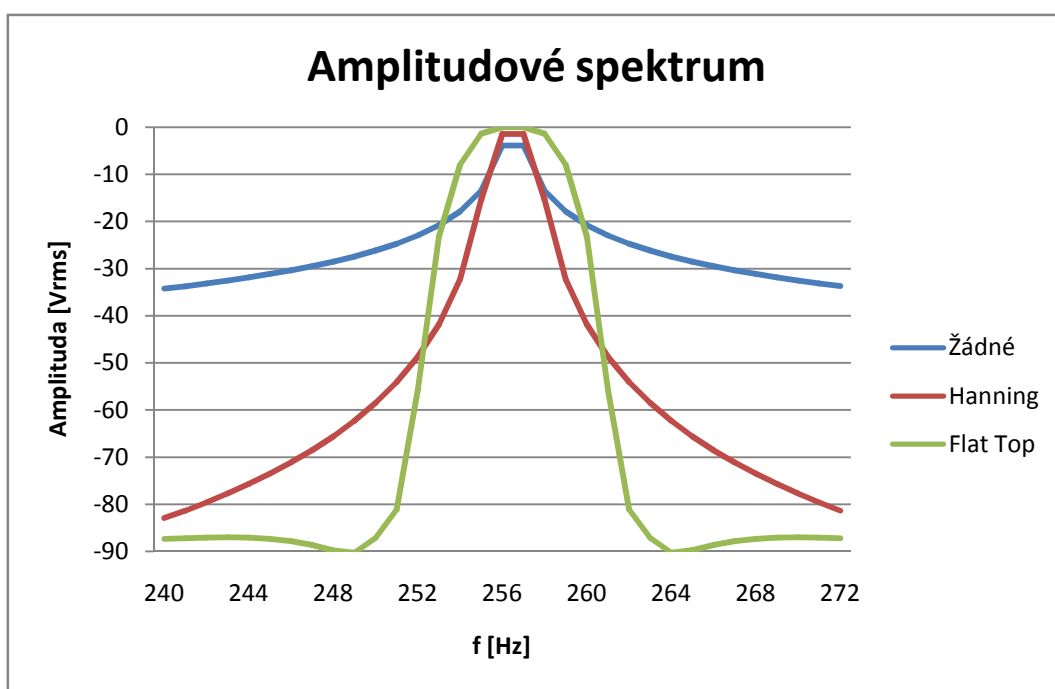
Signál na obr. 4.8 byl uměle vytvořen, jedná se o 256 period funkce sinus. Záznam má délku 1024 vzorků. Pokud záznam originálního signálu obsahuje celý počet period určité složky signálu, pak této složce odpovídá přesná frekvenční čára a je vidět pouze hlavní lalok spektra. Hlavní lalok je část frekvenční charakteristiky okénkovací funkce. Jednotkové obdélníkové okno má tento lalok nejužší, u Hanningova okna je širší a nejširší je u Flat Top okna. Při celém počtu period analyzovaného signálu dávají všechny okna stejnou magnitudu. A postranní laloky se neobjeví.

V tabulce 4.1 jsou uvedeny hodnoty amplitud pro frekvence okolo základní frekvence signálu. Jsou zde hodnoty frekvenčních čar od frekvence 254 Hz do 258 Hz



pro každý typ okénka. Chyba amplitudy pro frekvenci 256 Hz je 0 dB u každého typ okénka.

Pokud základní frekvence signálu není rovna násobku  $1/T$ , je spektrum signálu posunuto ze středu hlavního laloku o zlomek frekvence  $\Delta f$ . Ten odpovídá rozdílu mezi frekvencí signálu a frekvencí nejbližší frekvenční čáry Fourierovy transformace. Tento posun způsobí zobrazení postranních laloků a chybu amplitudy signálu. Obrázek 4.9 Obr. 4.9 Spektrum signálu s neceločíselným počtem period ukazuje rozptyl spektra signálu, který obsahuje 256,5 period.



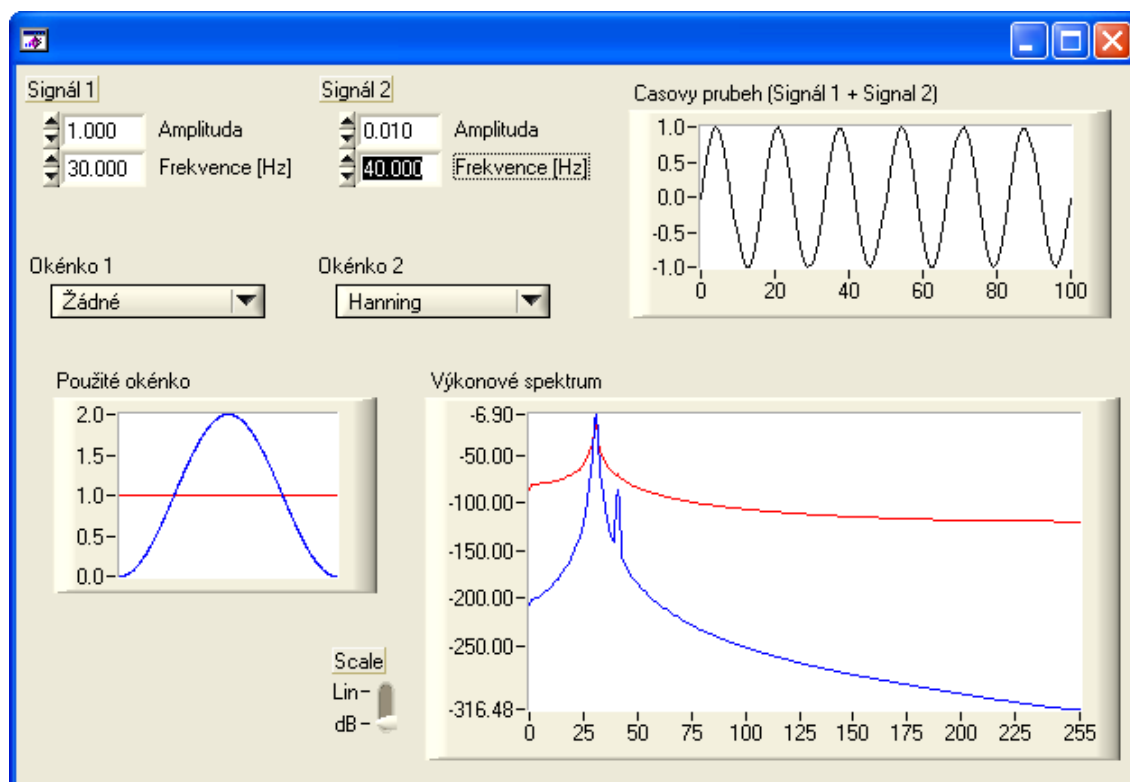
Obr. 4.9 Spektrum signálu s neceločíselným počtem period

Tab. 4.2 Hodnoty amplitud pro vybrané frekvence

Časové okno	254 [Hz]	255 [Hz]	256 [Hz]	257 [Hz]	258 [Hz]	Max. chyba amplitudy
Obdélníkové (žádné)	0,1273	0,2122	0,6366	0,6366	0,2122	0,3634
Hanning (Hann)	0,0243	0,1698	0,8488	0,8488	0,1698	0,1512
Flat Top	0,3988	0,8507	0,9990	0,9990	0,8507	0,0010

Na obrázku obr. 4.9 je spektrum signálu, který neobsahuje celý počet period. Je zde dobře vidět různý rozptyl spektra signálu při použití různých okének. Také je vidět vliv okének na chybu amplitudy. Flat top okénko má velmi malou chybu amplitudy, ale má více rozptýlené spektrum a vyšší postranní laloky než má Hanningovo okénko.

Obrázek 4.10 ukazuje frekvenční spektrum dvou signálů blízké frekvence o řádově různých amplitudách. Bez použití okénkovací funkce je rozptyl spektra prvního signálu tak velký, že převyšuje úroveň druhého signálu a tím činí druhý signál ve spektru neviditelný. Při použití Hanningova okénka je druhý signál ve spektru velice dobře patrný.



Obr. 4.10 Spektrum dvou signálů blízkých frekvencí

### 4.3 Vlastnosti okének

Fourierovu transformaci časových oken lze považovat za frekvenční charakteristiku pásmových filtrů (Tůma, 1997). Přehled vlastností okénkovacích funkcí se proto udává obdobným způsobem jako vlastnosti pásmových filtrů a je uveden v tabulce 4.3. Zvlnění (angl. Ripple) je stanoveno tak, že odpovídá frekvenčnímu pásmu o šířce  $1/T$  ve vzdálenosti od střední frekvence o  $\pm 0,5T$ . Toto pásmo totiž odpovídá ideálnímu filtru bez zvlnění. Důsledkem zvlnění je vznik amplitudové chyby při odečtu amplitud složek spektra.

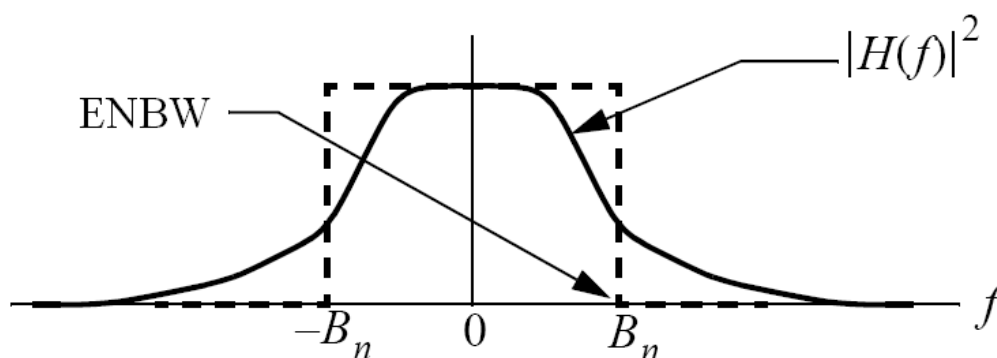
Tab. 4.3 Porovnání vlastností základních časových okének

Časové okno	Šířka pásma šumu	3dB pásmo	Ripple	Nejvyšší postranní lalok	Pokles laloků v dB/dek	60 dB pásmo	Tvarový faktor $B_{60dB}/B_{3dB}$
Obdélníkové	$\Delta f$	$0,089 \Delta f$	3,92dB	-13,3	20 dB	$665 \Delta f$	750
Hanning	$1,5 \Delta f$	$1,44 \Delta f$	1,42	-31,5	60 dB	$13,3 \Delta f$	9,2
Flat Top	$3,77 \Delta f$	$3,72 \Delta f$	0,01	-93,6	0 dB	$9,1 \Delta f$	2,5

### 4.3.1 Šířka pásma šumu

Šířka pásma šumu někdy také ENBW (z angl. Equivalent Noise Bandwidth) je jedním ze základních parametrů pásmových filtrů a používá se při popisu parametrů okénkovacích funkcí (Bruel & Kjaer, 1987).

Pro výpočet přenášeného výkonu signálu je důležitá efektivní šířka pásma šumu reálného filtru (okénka). Podle definice je efektivní šířka pásma šumu reálného filtru s frekvenční charakteristikou  $H(f)$  rovna šířce propustného pásma ekvivalentního ideálního filtru. Který na svůj výstup přenáší ze vstupního signálu typu bílý šum signál o shodném výkonu jako reálný filtr se stejným vstupním signálem. Efektivní šířka pásma se používá při výpočtu výkonu složek spektra ze signálu, který je vážen časovým oknem. Udává se v násobcích  $\Delta f$ . A je definována podle následujícího obrázku 4.11.



Obr. 4.11 Ekvivalentní šířka pásma šumu

## 4.4 Přehled typů okének

V této části je uveden přehled nejpoužívanějších typů okének podle (NI Windowing Signals). Jsou zde uvedeny jejich definiční vzorce, některé jejich významné vlastnosti a oblast jejich použití, pro kterou byly navrženy. U každého okénka je pak uveden graf, který se v programu objeví na stisk tlačítka. Tedy pokud má uživatel zájem vidět, jak vypadá tvar okénka.

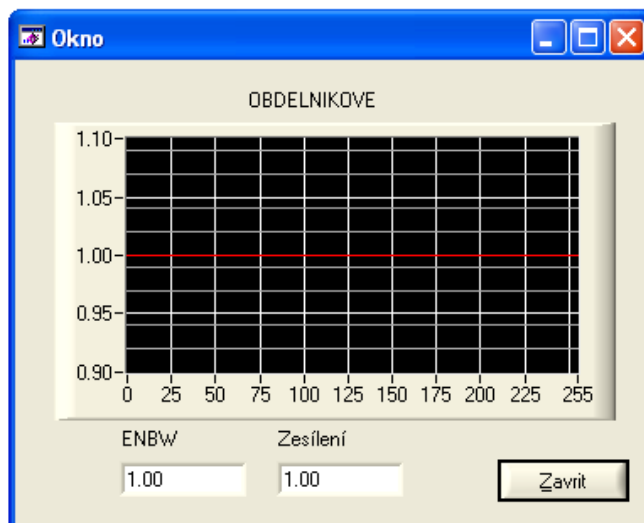
### 4.4.1 Obdélníkové (implicitní) okénko

Obdélníkové okno má hodnotu jedna po celé své délce, je definováno následujícím vztahem:

$$w(n) = 1 \text{ pro } n = 0, 1, 2, \dots, N - 1 \quad (4.4)$$

Kde  $N$  je délka okénka a  $w$  je jeho hodnota.

Použití Jednotkového obdélníkového okénka je stejné jako když nepoužijeme žádné. Obdélníková funkce jen ořízne signál na patřičnou délku. Obdélníkové okno má nejvíce rozptýlené spektrum.



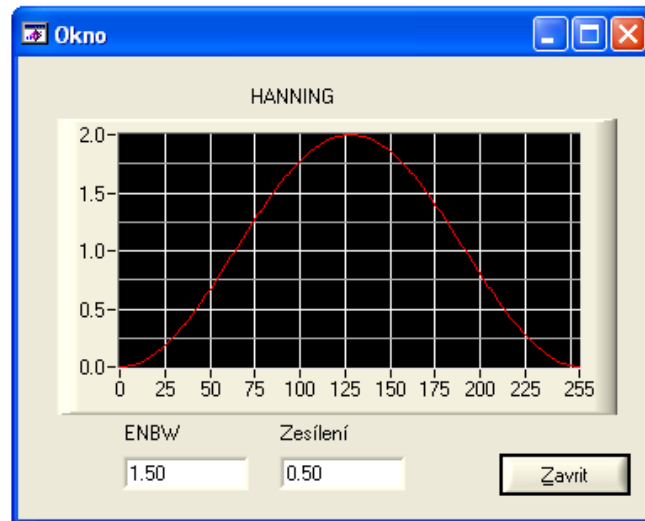
Obr. 4.12 Obdélníkové okénko

#### 4.4.2 Hanningovo okénko

Hanningovo okno má tvar podobný polovině periody funkce sinus. A je definováno následujícím vztahem:

$$w(n) = 1 - \cos \frac{2\pi n}{N} \text{ pro } n = 0, 1, 2, \dots, N - 1 \quad (4.5)$$

Kde  $N$  je délka okénka a  $w$  je jeho hodnota.



Obr. 4.13 Hanningovo okénko

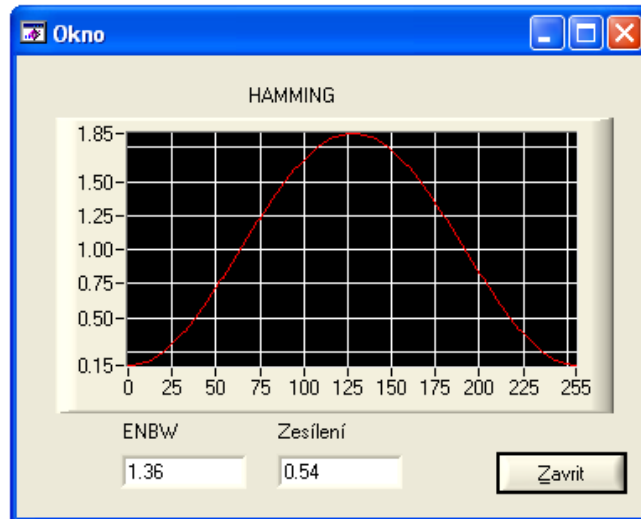
Hanningovo okénko je určeno pro všeobecné použití a pro analýzu přechodných signálů delších než je doba záznamu.

#### 4.4.3 Hamming okénko

Hammingovo je upravené Hanningovo okénko a je definováno následujícím vztahem.

$$w(n) = 0,977 - 0,855 \cos \frac{2\pi n}{N} \text{ pro } n = 0, 1, 2, \dots, N - 1 \quad (4.6)$$

Kde  $N$  je délka okénka a  $w$  je jeho hodnota.



Obr. 4.14 Hammingovo okénko

Hanningovo okno je velice podobné Hammingovu. Okna se navzájem se liší pouze tím, že Hammingovo okno nemá na krajích hodnoty tak blízko nule jako Hanningovo okno.

#### 4.4.4 Okénko Kaiser-Bessel

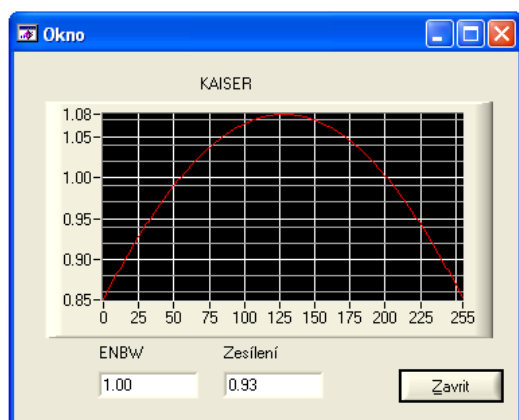
U okénka Kaiser-Bessel pomocí parametru **beta** může měnit jeho tvar. A tím i podle potřeby měnit rozptyl spektra. Je definováno následujícím vztahem.

$$w(n) = \frac{I_0\left(\beta \sqrt{1 - \left(\frac{2n}{N} - 1\right)^2}\right)}{I_0(\beta)} \quad \text{pro } n = 0, 1, 2, \dots, N - 1 \quad (4.7)$$

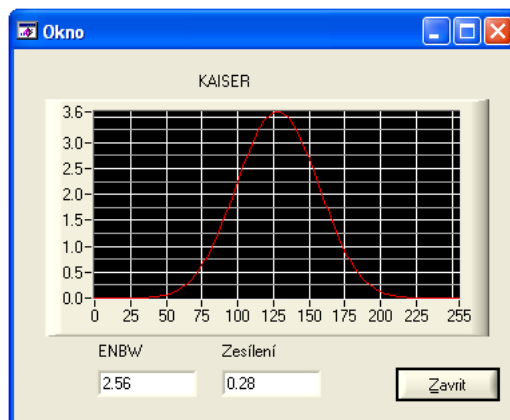
$$I_0(x) = \sum_{k=0}^{\infty} \frac{\left(\frac{x^2}{4}\right)^k}{(k!)^2} \quad (4.8)$$

Kde  $N$  je délka okénka,  $I_0$  je Besselova funkce nultého řádu a  $w$  je jeho hodnota.

Obrázky 4.15 a 4.19 ukazují různý tvar okénka v závislosti na parametru beta.



*Obr. 4.15 Okénko Kaiser-Bessel  
s parametrem Beta = 1*



*Obr. 4.16 Okénko Kaiser-Bessel  
s parametrem Beta = 20*

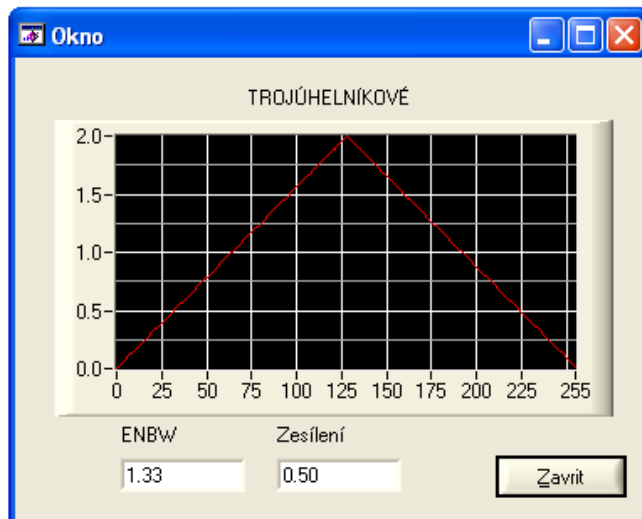
Čím je hodnota beta nižší, tím více se tvar okénka blíží tvaru obdélníku (pro hodnotu  $\beta = 0$  dostaneme obdélníkové okénko). S rostoucím hodnotou beta se mění tvar okénka do tvaru podobnému kuželu. Toto okno je nejvhodnější pro měření signálů o podobné frekvenci, ale s výrazně jinou amplitudou.

#### 4.4.5 Trojúhelníkové okno

Trojúhelníkové (někdy také Bartlettovo) okénko, má tvar trojúhelníku. Je definováno následujícím vztahem:

$$w(n) = 2\left(1 - \left|\frac{2n-N}{N}\right|\right) \text{ pro } n = 0, 1, 2, \dots, N-1 \quad (4.9)$$

Kde  $N$  je délka okénka a  $w$  je jeho hodnota.



Obr. 4.17 Trojúhelníkové okénko

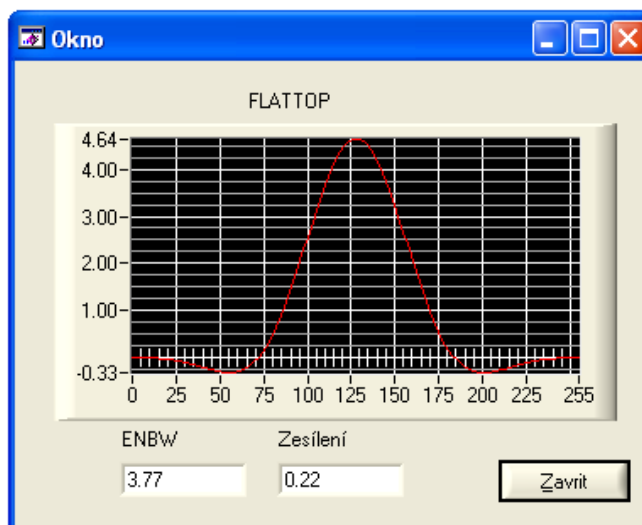
#### 4.4.6 Flat Top

Flat Top okénko má nejvyšší amplitudovou přesnost ze všech druhů okének. Pro signál ležící v polovině mezi dvěma diskrétními frekvencemi je to hodnota  $\pm 0.02$  dB. Ale Flat Top okénko má široký hlavní lalok a tím i špatné frekvenční rozlišení. Proto se nejčastěji využívá pro přesné měření amplitudy osamocených frekvenčních složek. Například při kalibraci. Okno je definováno následujícím vztahem.

$$w(n) = \sum_{k=0}^4 (-1)^k a_k \cos\left(k \frac{2\pi n}{N}\right) \text{ pro } n = 0, 1, 2, \dots, N-1 \quad (4.10)$$

$$a_0 = 1 \quad a_1 = -1,98 \quad a_2 = 1,29 \quad a_3 = -0,388 \quad a_4 = 0,0322$$

Kde  $N$  je délka okénka a  $w$  je jeho hodnota.



Obr. 4.18 Okénko Flat Top

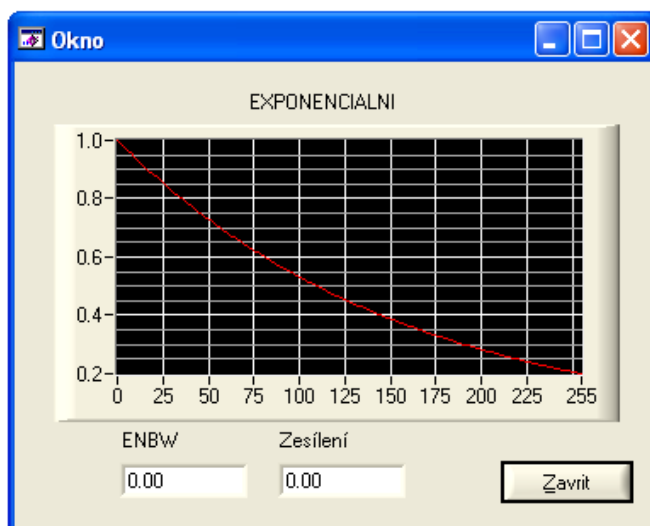


#### 4.4.7 Exponenciální okénko

Okénko má tvar klesající exponenciály a je dáno následujícím vztahem.

$$w(n) = e^{\left(\frac{n \cdot \ln(f)}{N-1}\right)} = f^{\left(\frac{n}{N-1}\right)} \quad \text{pro } n = 0, 1, 2, \dots, N-1 \quad (4.11)$$

Kde N je délka okénka, je konečná hodnota f z intervalu od 0 do 1.



Obr. 4.19 Exponenciální okénko s koncovou hodnotou 0,2

Okénko se používá pro analýzu přechodných signálů, jejichž délka je delší než časové okno. Okénko tlumí konec signálu, tím zajistí to, že je signál na konci záznamu plně utlumen.

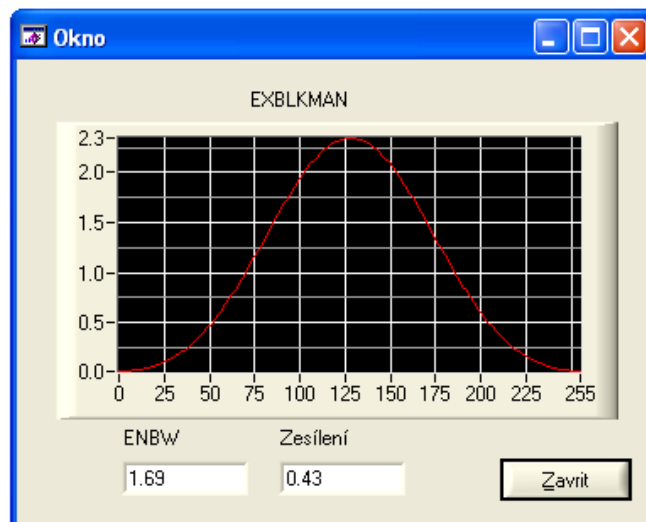
#### 4.4.8 Okénko Exact Blackman

Používá se pro měření samostatných složek signálů. Okno Exact Blackman má užší hlavní lalok a nižší maximální postranní lalok než okno Blackman, ale okno Blackman má rychlejší pokles postranních laloků. Okénko je definováno následujícím vztahem:

$$w(n) = \frac{7938}{18608} - \frac{9240}{18608} \cos\left(\frac{2\pi n}{N}\right) + \frac{1430}{18608} \cos\left(2 \frac{2\pi n}{N}\right) \quad (4.12)$$

$$\text{pro } n = 0, 1, 2, \dots, N-1$$

Kde N je délka okénka a w je jeho hodnota.



Obr. 4.20 Okénko Exact Blackman

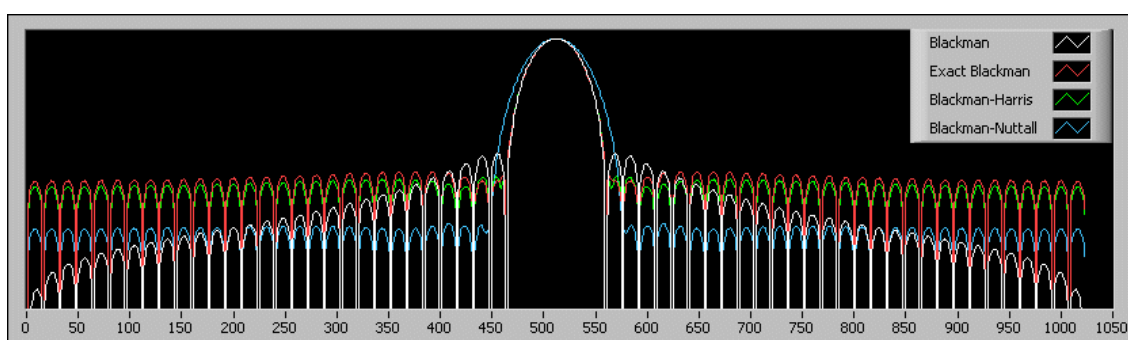
#### 4.4.9 Blackman, Blackman-Harris, Blackman-Nuttall

Jedná se pouze o upravené verze okénka Exact Blackman. Například okénko Blackman se liší pouze tím, že má v definičním vzorci koeficienty zaokrouhlené na dvě desetinná místa. Okénko Blackman Nuttall má přidáný jeden koeficient.

$$w(n) = 0,42 - 0,50 \cos\left(\frac{2\pi n}{N}\right) + 0,08 \cos\left(2 \frac{2\pi n}{N}\right) \quad (4.13)$$

$$\text{pro } n = 0, 1, 2, \dots, N - 1$$

Definiční vzorec okénka Blackman, kde  $N$  je délka okénka a  $w$  je jeho hodnota.



Obr. 4.21 Spektra okének typu Blackman

Na předchozím obrázku 4.20 je frekvenční spektrum všech okének typu Blackman. Je zde vidět, že okénko Blackman má nejužší hlavní lalok a největší rychlost poklesu postranních laloků. Okénko Blackman-Nuttall má zase nejširší hlavní lalok a nejnižší maximální výšku postranního laloku.

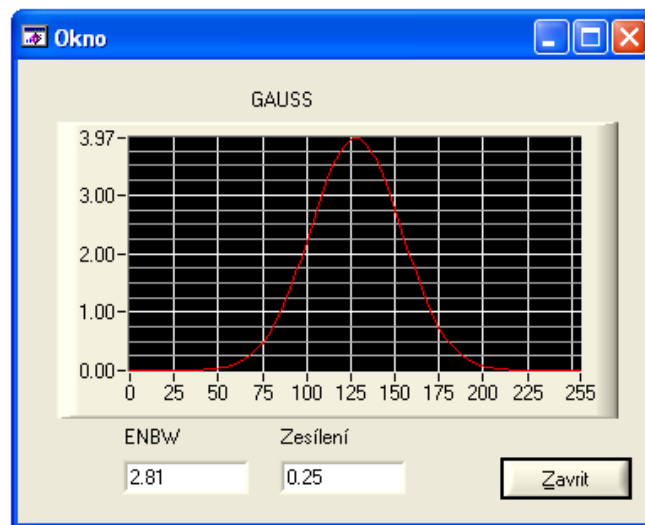
#### 4.4.10 Gaussovo okénko

Gaussovo okénko je definováno následujícím vztahem:

$$w(n) = e^{-\frac{(n-m)^2}{2(\sigma N)^2}} \quad (4.14)$$

$$\text{pro } n = 0, 1, 2, \dots, N - 1$$

Kde  $N$  je délka okénka,  $m=(N-1)/2$ ,  $\sigma$  je směrodatná odchylka a  $w$  je hodnota okénka v bodě  $n$ .



Obr. 4.22 Gaussovo okénko

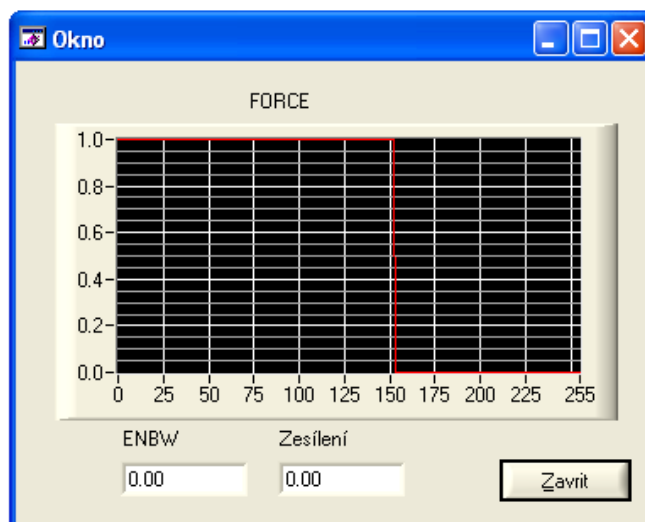
#### 4.4.11 Okénko Force

Okénko se používá k analýze přechodových signálů a je definováno následujícím vztahem.

$$w(n) = 1 \text{ pro } 0 \leq n \leq d \quad (4.15)$$

$$w(n) = 0 \text{ pro hodnoty } d < n < N$$

Kde  $N$  je délka okénka,  $w$  je jeho hodnota a parametr  $d$  udává počet vzorků, kdy je  $w$  nenulové.



Obr. 4.23 Okénko Force s parametrem  $d=153$

#### 4.4.12 Další typy okének

Ještě existuje několik dalších typů okének, jako je obecné kosinusové, zúžené kosinusové nebo okénko Dolph-Chebyshev, které umožňuje pomocí parametru nastavit hladinu postranních laloků vůči hlavnímu laloku.

### 4.5 Strategie pro výběr okének

Pro různé aplikace se používají různá okna, každé okno má různé charakteristické vlastnosti, které jsou vhodné pro různé aplikace. Ke správnému výběru časového okna musíme dopředu znát nebo odhadnout povahu měřeného signálu (NI Windowing Signals). Pokud jsou v měřeném signálu obsaženy silné rušivé složky a jejich frekvence se výrazně liší od signálu, který nás zajímá, tak se volí okénka s velkým poklesem postranních laloků. Ale pokud je frekvence rušivých složek blízká frekvenci zkoumaného signálu, tak volíme okénko s nízkou maximální výškou postranních laloků.

V případě, že potřebujeme změřit dva signály s podobnou frekvencí, doporučuje se volit okénko s úzkým hlavním lalokem. Pokud nám jde o zjištění přesné amplitudy signálu a už nás tolik nezajímá jeho přesná frekvence, volíme okénko se širokým hlavním lalokem. Implicitní jednotkové okénko je vhodné pro signály, které obsahují jen složky o frekvencích násobku  $1/T$  nebo pro signály, které obsahují široké spektrum frekvencí.

Pro velkou většinu aplikací vyhovuje Hanningovo okénko, které má dobré

frekvenční rozlišení a zmenšuje rozptyl spektra. Flat Top okénko zase má velmi malou chybu amplitudy, ale jeho široký hlavní lalok způsobuje špatné frekvenční rozlišení a rozptyl spektra. Flat Top okénko má také nižší maximální výšku postranního laloku. Hanningovo okénko má zase rychlejší pokles postranních laloků. Pokud analyzujeme signál neznámé povahy, je nejlepší použít Hanningovo okénko. Na obrázcích 4.8 a 4.9 jsou vidět rozdílné vlastnosti základních okének v praxi.

Pokud analyzujeme přechodné signály a impulzní odezvy, není vhodné používat klasické okénkovací funkce, protože potlačují důležité informace na začátku záznamu. Místo nich se používají Exponenciální okénka a okénka Force. Exponenciální okénka jsou vhodná pro analýzu impulzní odezvy, protože pozvolně utlumují signál, až k nule na konci záznamu. Okénka Force se používají k analýze rázově buzených signálů, protože odstraňují nežádoucí signály na konci záznamu.

Výběr správné okénkovací funkce není vždy snadný a jednoznačný, protože neexistuje žádný univerzální postup pro výběr správného okénka. Počáteční výběr může usnadnit následující tabulka.

*Tab. 4.4 Typ okénka podle signálu*

Povaha signálu	Okénko
Sinusový signál a jeho kombinace	Hanning
Sinusový signál (přesná amplituda)	Flat Top
Úzko-pásmový náhodný signál (měření vibrací)	Hanning
Širokopásmový náhodný signál (bílý šum)	Obdélníkové (žádné)
Sinusové signály o blízkých frekvencích	Obdélníkové, Hanning
Vybuzené (úderem kladívka)	Force

## 5 Zvukový vstup a výstup

### 5.1 Ověření vlastností zvukové karty

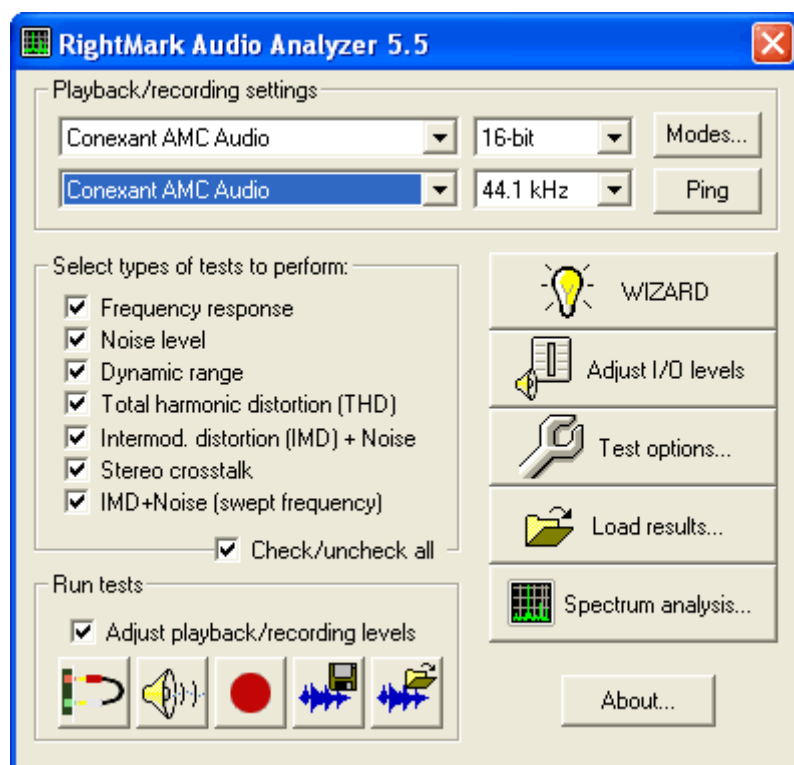
Pro ověření vlastností zvukové karty a její vhodnosti pro tento výukový program byl použit freewarový program RightMark Audio Analyzer. Ten je k dispozici ke stažení zdarma na adrese [audio.rightmark.org](http://audio.rightmark.org). Jediné co tedy k otestování zvukové karty potřebujeme, je již zmíněný software a mikrofonní kabel, tedy kabel který má na obou koncích koncovku 3,5mm Jack. Tímto kabelem propojíme linkový výstup (Line-out, Rear) se vstupem Line-in nebo se vstupem pro mikrofon<sup>2</sup> a můžeme začít měřit. Tato metoda měření (propojení vstupu a výstupu) nepatří mezi nejpřesnější, protože nelze určit, jestli je chyba na vstupní či výstupní části karty. Ale její výhodou je, že si ji může vyzkoušet každý uživatel sám na svém počítači a své zvukové kartě. Nemělo by smysl změřit jednu konkrétní zvukovou kartu, když program bude používán na jiných počítačích s různými zvukovými kartami. Tyto karty mohou mít diametrálně rozdílné vlastnosti. Tato metoda dá uživateli pouze představu, co je s takovou kartou schopen měřit. A případně pokud by měl zájem tak mu poskytne návod jak podobné měření provést pomocí profesionálního signálového analyzátoru nebo referenční zvukové karty.

#### 5.1.1 Co je RMAA a jak pracuje

RightMark Audio Analyzer dále jen RMAA je jednoúčelovým programem k testování zvukových karet. Dokáže generovat testovací signál a posílat ho na výstup zvukové karty. Ten je pak propojovacím kabelem přiveden zpět na vstup karty a následně analyzován.

---

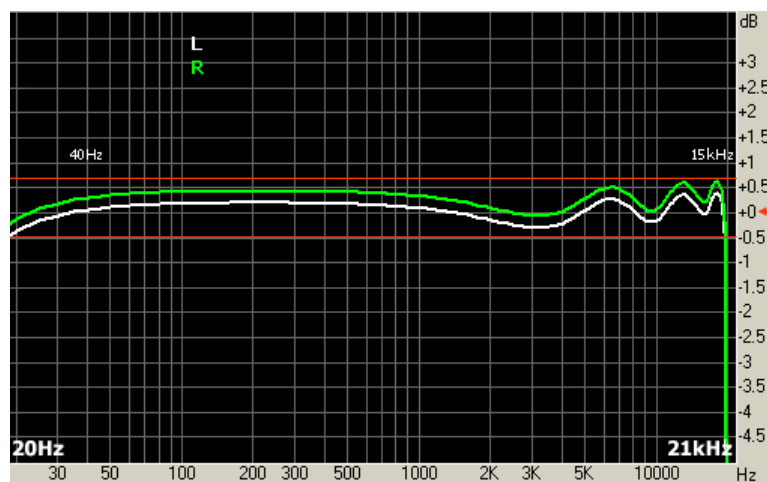
<sup>2</sup> Mikrofonní vstup se nedoporučuje používat, je většinou monofonní a bývá vybavený předřadným zesilovačem s funkcí "Gain control", který dynamicky mění úroveň signálu.



obr. 5.1 Základní obrazovka programu RMAA

### 5.1.2 Co program naměří

Pro nás je nejdůležitější funkcí měření frekvenční charakteristiky karty. Program charakteristiku znázorní do grafu a odečte, zní maximum a minimum. Z této charakteristiky pak snadno zjistíme, v jakém rozsahu frekvencí je karta použitelná. Dále program vypočítává hladinu šumu, dynamický rozsah, harmonické zkreslení, intermodulační zkreslení a přeslech mezi kanály.



obr. 5.2 Frekvenční charakteristika karty

## 5.2 API funkce pro ovládání multimédií ve Windows

Aby mohl program nahrávat a přehrávat zvuky, využívá předem připravené funkce operačního systému. Tak zvané API funkce.

### 5.2.1 Co to je API?

API, neboli aplikační programovací rozhraní (application programming interface), je programovací rozhraní vyvinuté Microsoftem pro programování ve Windows v jazyce C/C++, ale nejen v něm. Operační systém Windows má definované své API, které obsahuje všechna volání funkcí, které může aplikace volat z operačního systému. Také obsahuje definice souvisejících datových typů a struktur.

Z pohledu programátorů je Win32 API programovací rozhraní, které slouží pro komunikaci s operačním systémem Windows. Umožňuje využívat jeho různé funkce, datové typy a struktury pro tvorbu aplikací založených na grafickém rozhraní.

Jedná se o tisíce funkcí, které nám umožní provádět různé akce, aniž bychom ty funkce museli vymýšlet, používat datové typy a struktury OS Windows. Win API definuje celou škálu funkcí pro grafiku (GDI), zvuk, ovládací zařízení, síť, aj.

## 5.3 Práce s multimédií

API funkce pro práci s multimédií se dají rozdělit do dvou úrovní (Petzold, 1998). Na funkce nízké úrovně tak zvané „low-level“ a funkce vysoké úrovně tedy „high-level“. Nízko-úrovňové funkce pro ovládání zvukového vstupu a výstupu začínají prefixem *wavein* resp. *waveout*.

Vysoko-úrovňové rozhraní pro práci s multimédií se nezývá MCI (Media Control Interface). Toto rozhraní umožňuje ovládat všechna multimediální zařízení v počítači. Ovládání všech zařízení se provádí podle jednoduchého modelu. Otevře se zařízení pro vstup nebo výstup. Spustí se nahrávání nebo přehrávání a na konec zařízení zavřeme.

Hlavní rozdíl mezi vysokoúrovňovým MCI rozhraním a nízko-úrovňovým rozhraním je ten, že rozhraní MCI pracuje přímo se soubory. Například ukládá nahrávaný zvuk přímo do wave souboru a naopak při přehrávání zvuk ze souboru čte. Tento přístup nám ve většině případů oproti nízko-úrovňovému rozhraní ušetří hodně práce. Na druhou stranu je toto řešení méně flexibilní.



Další možností jak přistupovat k multimediálním zařízením je použití DirectX API, ale jeho popis je nad rámec této práce.

MCI jde používat dvěma různými, ale navzájem podobnými způsoby. První způsob ovládání používá posílání zpráv. Tyto zprávy jsou obdobné jako běžné zprávy ve Windows. Tyto zprávy obsahují bitové příznaky a datové struktury. K posílání těchto zpráv se používá se funkce `mciSendCommand`.

```
MCIERROR mciSendCommand(  
    MCIDEVICEID IDDevice,  
    UINT         uMsg,  
    DWORD        fdwCommand,  
    DWORD        dwParam,  
);
```

Která má parametry identifikátor zařízení, které zprávu obdrží. Dále typ zprávy, příznaky zprávy a ukazatel na strukturu, která obsahuje parametry zprávy. Parametry mohou být například požadovaný formát wave souboru který se má nahrát.

Druhý způsob používání MCI rozhraní je posílání textových ASCII řetězců. Tyto řetězce se posílají pomocí funkce `mciSendString`.

```
MCIERROR mciSendString(  
    LPCTSTR lpszCommand,  
    LPTSTR  lpszReturnString,  
    UINT    cchReturn,  
    HANDLE  hwndCallback,  
);
```

Tato funkce má jako první parametr ukazatel na nulou ukončený řetězec s MCI příkazem. Dále pak ukazatel na buffer kde čekáme odpověď, velikost tohoto bufferu a nakonec handle callback okna.

## 6 Přehled formátu wave

Wave formát je základní formát souborů pro uložení audio dat v systému Windows. Díky popularitě tohoto systému a velkému množství programů napsaných pro tuto platformu se stal jedním z nejvíce podporovaných audio formátů na počítačích typu PC. To, co tento formát činí nejvíce užitečným, je fakt, že téměř každý program ve Windows, který umí otevřít nebo uložit digitální záznam zvuku, podporuje tento formát souborů.

### 6.1 Formát dat

Wave soubory jsou přirozeným formátem audio souborů pro operační systém Windows, založeném na architektuře procesorů typu x86. Všechna data uložená tímto systémem jsou uložena v tzv. „Little-endian“ pořadí. Kdy je v paměti uložen nejméně významný bajt jako první.

Příklad uložení hodnoty 0x0A0B0C0D v paměti:

- *8bitů velkým základním elementem paměti:*

adresa→					
...	0x0D	0x0C	0x0B	0x0A	...

Nejméně významný byte (*LSB*) 0x0D je uložen na nejnižší adrese. Ostatní bajty následují v pořadí, podle jejich stoupající významnosti.

- *16-bitovým základním elementem paměti:*

adresa→			
...	0x0C0D	0x0A0B	...

V prvních 16bitech je uložena hodnota s nižším významem 0x0C0D, následovaná hodnotou 0x0A0B.

#### 6.1.1 Řetězce

Wave soubory obsahují textové řetězce pro označení částí souborů, vytváření poznámek, atd. Řetězce jsou uloženy ve stejném formátu jako ten, který používá jazyk Pascal. Tedy první byte určuje délku řetězce (počet ASCII znaků). Na dalších bytech jsou uloženy jednotlivé ASCII znaky textu.

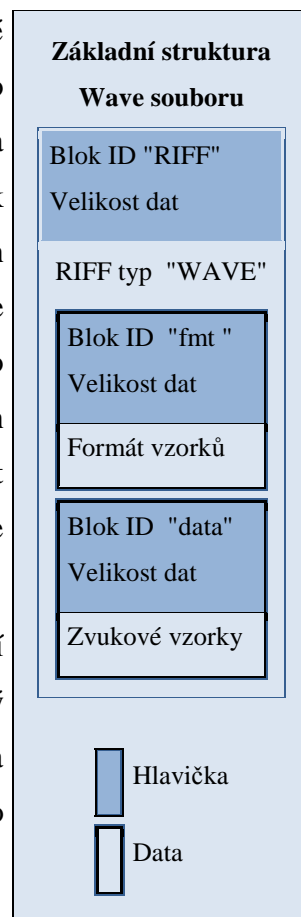
Tab. 6.1 Příklad řetězce

7	'p'	'r'	'i'	'k'	'l'	'a'	'd'
---	-----	-----	-----	-----	-----	-----	-----

## 6.2 Struktura souboru

Wave soubory podle (Wilson, 2003) používají standardní RIFF (Resource Interchange File Format) strukturu. Ta se také používá například pro ukládání multimediálních souborů, jako jsou AVI<sup>3</sup> nebo RMI<sup>4</sup> soubory. Tato struktura dělí soubory na bloky zvané: „Chunk“, což se dá přeložit jako poleno. Každý blok má vlastní hlavičku a vlastní data. Hlavička určuje typ uložených dat a jejich velikost v bytech. Toto uspořádání souboru, umožňuje programům které nepoužívají nebo neznají určitý druh těchto bloků, tyto bloky přeskočit a pokračovat zpracováváním jiných bloků, které program již zná. Některé bloky mohou obsahovat vnořené bloky, tzv. „sub-chunks“. Na obrázku vpravo je vidět, že blok „RIFF“ obsahuje bloky „fmt“ a „data“.

Velikost jednotlivých bloků RIFF souboru v bytech musí být násobkem dvou. To znamená, že když blok obsahuje lichý počet datových bytů, musí být doplněno nulovým bajtem na sudou délku. Tento přidání bajt se však nepočítá do údaje o velikosti bloku, který je uveden v hlavičce.



<sup>3</sup> AVI je formát společnosti Microsoft pro uložení audio a video dat

<sup>4</sup> RMI je formát pro uložení hudebních MIDI souborů

## 6.3 Hlavička Wave souboru

V následující tabulce 6.2 je uvedena celá hlavička wave souborů. V tabulce jsou uvedeny velikosti jednotlivých částí, jejich umístění a způsob jejich uložení v paměti.

Tab. 6.2 Hlavička wav souboru

endian	Pozice	Jméno pole	Velikost	
big	0	ID Bloku	4	Základní blok „RIFF“, v položce formát je uveden řetězec „wave“. Ten vyžaduje bloky „fmt“ a „data“.
little	4	Velikost bloku	4	
big	8	Formát	4	
big	12	ID vnořeného bloku 1	4	Vnořený blok „fmt“, popisuje formát dat uložených v bloku „data“.
little	16	Velikost vnořeného bloku 1	4	
little	20	Komprese	2	
little	22	Počet kanálů	2	
little	24	Vzorkovací frekvence	4	
little	28	Datový tok	4	
little	32	Velikost vzorku	2	
little	34	Počet bitu na vzorek	2	
big	36	ID vnořeného bloku 2	4	
little	40	Velikost vnořeného bloku 2	4	
little	44	data	Velikost vnořeného bloku 2	
				Vnořený blok „data“, obsahuje jednotlivé vzorky.

### 6.3.1 Blok RIFF

Hlavička wave souboru je standardní hlavičkou souborů typu RIFF. Prvních 8 bajtů obsahuje identifikátor bloku („RIFF“) a jeho velikost. Tato velikost se rovná velikosti celého souboru minus osm bajtů určených pro hlavičku. První čtyři datové bajty bloku RIFF určují jeho typ. Wave soubory zde obsahují řetězec „WAVE“. Dále se v souboru nacházejí bloky, které určují formát zvukových vzorků nebo různé efekty jako jsou např.: smyčky nebo dozvuky. Nakonec, se zde nachází blok s vlastními vzorky.

Tab. 6.3 Blok RIFF

Pozice	Velikost	Popis	Hodnota
0x00	4	Druh bloku	"RIFF" (0x52494646)
0x04	4	Velikost dat	(Velikost souboru) - 8
0x08	4	RIFF typ	"WAVE" (0x57415645)
0x10	Bloky Wave		

## 6.4 Bloky Wave souboru

Wave soubory vždy obsahují dva základní bloky. Je to blok Formát a blok Data. Blok Formát obsahuje údaje o vzorcích a způsobu jejich uložení. Blok Data obsahuje samotné vzorky signálu. Ačkoliv to není ve specifikaci Wave formátu vyžadováno, je z praktických důvodů vhodnější umístit blok Formát před blok Data. Tento způsob je snadnější na zpracování a také mnoho programů toto pořadí předpokládá. Při jeho nedodržení by mohlo dojít k chybnému zpracování souboru.

Všechny RIFF bloky a tudíž i WAVE bloky jsou uloženy v následujícím formátu.

*Tab. 6.4 Uspořádání RIFF bloků*

Pozice	Velikost	Popis
0x00	4	Identifikátor bloku
0x04	4	Velikost dat
0x08	Data	

### 6.4.1 Blok Formát - "fmt "

Blok formát obsahuje informaci o tom, jak jsou data uložena a jak by měla být přehrána. To zahrnuje způsob uložení, vzorkovací frekvenci, počet zvukových kanálů, komprese, počet bitů na jeden vzorek a další.

*Tab. 6.5 Uspořádání bloku Formát*

Pozice	Velikost	Popis	Hodnota
0x00	4	Identifikátor bloku	"fmt " (0x666D7420)
0x04	4	Velikost dat	16 + rozšiřující bajty
0x08	2	Kód komprese	1 - 65,535
0x0a	2	Počet kanálů	1 - 65,535
0x0c	4	Vzorkovací frekvence	1 - 0xFFFFFFFF
0x10	4	Datový tok za 1s	1 - 0xFFFFFFFF
0x14	2	Velikost vzorku	1 - 65,535
0x16	2	Počet bitů na vzorek	2 - 65,535
0x18	2	Počet rozšiřujících bajtů	0 - 65,535
0x1a	Rozšiřující bajty		

### Identifikátor bloku a velikost dat

Identifikátor bloku je vždy "fmt " (0x666D7420) a velikost standardního bloku Formát je 16B plus počet rozšiřujících bajtů pro dekompresi dat. Tyto doplňkové bajty jsou použity pouze v případě komprese dat. Je dobré zmínit, že identifikátor "fmt " končí znakem mezery (0x20).

### Kód komprese dat

První dva datové bajty bloku Formát určují způsob komprese dat. V následující tabulce je uveden seznam dnes nejpoužívanějších kódů.

Tab. 6.6 Kompresní kódy Wave souborů

Kód	Popis
0 (0x0000)	Neznámý
1 (0x0001)	PCM (nekomprimováno)
2 (0x0002)	Microsoft ADPCM
6 (0x0006)	ITU G.711 a-law
7 (0x0007)	ITU G.711 $\mu$ -law
17 (0x0011)	IMA ADPCM
20 (0x0016)	ITU G.723 ADPCM (Yamaha)
49 (0x0031)	GSM 6.10
64 (0x0040)	ITU G.721 ADPCM
80 (0x0050)	MPEG

### Počet kanálů

Položka počet kanálu určuje, kolik zvukových stop je uloženo v bloku Data. Hodnota jedna znamená mono signál, hodnota dvě znamená stereo signál, atd.

### Vzorkovací frekvence

Udává počet vzorků za sekundu, pořízených na jeden zvukový kanál.

### Datový tok za jednu sekundu

Hodnota udává počet bajtů za sekundu, který musí být poslán na D/A převodní zvukové karty, aby byl soubor přehrán. Ta hodnota je dána vzorkovací frekvencí vynásobenou velikostí jednoho zvukového vzorku a počtem kanálů.

$$\text{datový tok} = F_s * \text{Počet\_kanálů} * \text{Poč\_bit\_na\_vzorek} / 8$$

### Velikost vzorku

Hodnota má význam velikosti jedné datové jednotky v bajtech pro jeden vzorek a všechny použité kanály.

$$\text{velikost vzorku} = \text{Počet\_kanálů} * \text{Počet\_bitů\_vzorku} / 8$$

### Počet bitů na vzorek

Toto číslo udává počet bitů jednoho vzorku jednoho kanálu, ve kterých je uložena informace. Obvykle nabývá hodnoty 8, 16, 24 nebo 32. Není-li počet bitů násobkem čísla 8, je zaokrouhlen na nejbližší vyšší násobek osmi a doplněn nulami. Tyto bity jsou při zpracování vynechány.

### Počet rozšiřujících bajtů

Hodnota určuje, kolik doplňujících bajtů následuje. Tato hodnota se uvádí, pouze když je hodnota komprese jiná než 0x0001, tedy PCM. Uvádí se u souborů, kde je použita komprese dat a je potřebná nějaká dodatečná informace k jejich dekódování. Není-li velikost této dodatečné informace násobkem dvou bajtů, je doplněna nulami. Údaj o velikosti se však nezmění.

## 6.4.2 Blok Data - "data"

Blok Wave Data obsahuje digitální audio data, která lze načíst a dekódovat pomocí informací uložených v bloku Formát. Pokud je kód komprese 1, tedy nekomprimovaná PCM, tak soubor obsahuje surové nevzorkované hodnoty. Wave soubory obvykle obsahují pouze jeden blok data, ale mohou jich obsahovat i více.

Tab. 6.7 Uspořádání bloku data

Offset	Délka	Typ	Popis	Hodnota
0x00	4	char[4]	Identifikátor bloku	"data" (0x64617461)
0x04	4	dword	Velikost bloku	Záleží na velikosti vzorku a kompresi
0x08	Data			

Vícekanálové audio signály mají vzorky uložené prokládaně. To znamená, že uložené vzorky vícekanálového audiosignálu<sup>5</sup> se cyklicky střídají po celé délce záznamu. To je výhodné hlavně při práci s velkými soubory a dovoluje nám to mít v paměti načtenou pouze část souboru. V tabulce 6.8 je uveden příklad, jak je uložen dvoukanálový signál ve wave souboru.

<sup>5</sup> Například stereo signálu nebo vícekanálového prostorového signálu

Tab. 6.8 Uložení dvoukanálové zvuku ve wave souboru

Vzorek	Kanál	Hodnota
0	1 (left)	0x0053
	2 (right)	0x0024
1	1 (left)	0x0057
	2 (right)	0x0029
2	1 (left)	0x0063
	2 (right)	0x003C

Ještě dvě důležité poznámky na konec. Vzorky uložené jako 8 bitová čísla jsou specifikovány jako neznaménkové. Jiné velikosti vzorků jsou specifikovány jako znaménkové. Například šestnácti bitové vzorky mohou nabývat hodnot od -32,768 do +32,767. Jak již bylo zmíněno dříve, všechny RIFF bloky (včetně bloku data) musí být zarovnané. Pokud máme lichý počet vzorků, doplníme ho na konci o nulový bajt.

### 6.4.3 Další možné bloky wave souborů

Ve wave souborech se používají ještě další typy bloků. Přehled těch nejpoužívanějších a nejznámějších je uveden v následující tabulce.

Blok	Identifikátor	Popis
<b>Fakt</b>	"fact"	Obsahuje informace závislé na způsobu komprese dat.
<b>Seznam</b>	"wavl"	Pořadí střídání bloků "data" a "slnt".
<b>Ticho</b>	"slnt"	Pro zmenšení velikosti souboru. Udává počet vzorku, kdy je ticho.
<b>Záložka</b>	"cue "	K označení míst v záznamu, jako začátek sloky, atd.
<b>Playlist</b>	"plst"	Pořadí záložek pro přehrání souboru.

## 6.5 Odchyly od formátu

Jak je napsáno na (Sonic), díky velké popularitě formátu wave existují stovky programů, které tento formát podporují. V tak velkém množství programů se najde několik, které vytvářejí ne zcela správné soubory. A už je to z důvodu chyby programátora nebo špatné specifikace. Takovéto programy mohou být poměrně rozšířené a vytvářet miliony nekorektních souborů. Proto je nutné, aby byl program schopen se s těmito soubory vypořádat, případně na jejich chybu upozornit.

- Špatný údaj o velikosti vzorku. Tuto chybu lze opravit výpočtem z dříve uvedeného vztahu.
- Špatný údaj o velikosti datového bloku. Tuto chybu lze také opravit výpočtem z dříve uvedeného vztahu.



- Nedoplnění na sudý počet bajtů. Řešením je dát uživateli varování, když je nalezen neznámý identifikátor bloku. Posunem o jeden bajt lze data přečíst.

## 7 Popis vlastní aplikace

Aplikace je napsána v jazyce ANSI C ve vývojovém prostředí LabWindows CVI. Toto vývojové prostředí je primárně určeno pro vytváření inženýrských aplikací, jako jsou virtuální zařízení na analýzu a zpracování dat, nebo pro vytváření řídicích, kontrolních a testovacích aplikací. K vývoji takovýchto aplikací je v prostředí k dispozici množství speciálních funkcí a knihoven. A právě velké množství funkcí pro analýzu signálů bylo také hlavním důvodem pro výběr tohoto vývojového prostředí. Hlavním záporem tohoto prostředí je jazyk ANSI C, který neumožňuje objektově orientované programování (OOP) a tudíž méně vhodný pro vytváření uživatelských aplikací.

### 7.1 Členění zdrojového kódu programu

Při tvorbě rozsáhlejších projektů je vhodné zdrojový kód rozdělit do více částí. Tím se zvýší přehlednost kódu a urychlí se jeho překlad. Každá takováto část by pak měla sdružovat funkce určitého typu, například funkce pro práci s grafikou nebo se soubory. Výhodou je, že každá takováto část je víceméně nezávislá a zvyšuje přehlednost při vytváření programu.

Jazyk C umožňuje práci s tzv. moduly, ty oproti prostému vkládání přinášejí mnohé výhody. Jednotlivé soubory jsou přeloženy samostatně. Po překladu jsou vzniklé soubory (moduly) spojeny do jednoho programu tzv. *linkerem*. Výhody tohoto postupu jsou zřejmé. Při dalších překladech jsou již kompilovány pouze ty soubory, ve kterých skutečně došlo ke změně, což celý proces značně urychluje. Funkce i proměnné z různých modulů mohou i nemusí být sdíleny, a je tedy možné mít v různých modulech definovány různé funkce se stejným jménem. Další výhodou je pak možnost připojovat a používat i moduly vytvořené v jiném programovacím jazyku.

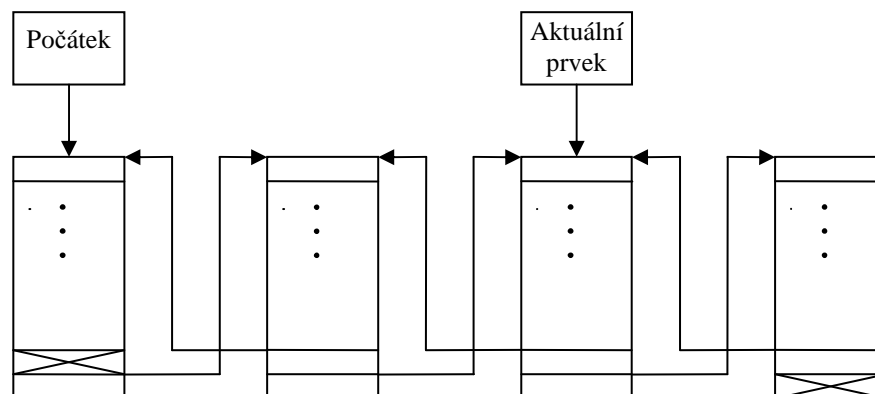
Mezi moduly lze sdílet pouze globální datové objekty, neboť ty existují po celou dobu vykonávání programu. Nepřímo pak, voláním sdílené funkce, lze sdílet i statické lokální proměnné. Lokální automatické proměnné sdílet nelze.

Sdílená proměnná musí být definována právě v jednom modulu. V ostatních modulech, ve kterých chceme k této proměnné přistupovat, musí být tato proměnná již pouze deklarována. Tedy již nedochází k dalšímu přidělení paměti, pouze je překladači oznámena její existence v jiném modulu.

Deklarace sdílené proměnné vypadá stejně jako její definice, jen s tím rozdílem, že před ní ještě uvedeme klíčové slovo *extern*. V deklaraci proměnné nikdy nesmíme použít její inicializaci. Překladač by pak deklaraci považoval za definici bez ohledu na klíčové slovo *extern*.

Z výše zmíněných důvodů byl program rozdělen do čtyř částí. Rozdělení je provedeno podle účelu, pro který jsou funkce v programu použity. Jednotlivé moduly jsou do sebe postupně vnořeny a vytvářejí tak stromovou strukturu. V hlavním modu je vytvořeno uživatelské rozhraní programu a jsou zde funkce, které se starají o obsluhu událostí, ze kterých se volají funkce z ostatních modulů programu. Do tohoto modulu je vnořený modul pro práci se zvukovou kartou a pro načítání a ukládání souborů. Ačkoli se může zdát, že přehrání zvuku a uložení signálu souboru jsou dvě podstatně rozdílné funkce, díky knihovně MCI (z angl. Media Control Interface) se prakticky jedná o totožnou operaci. Signál se uloží do souboru a pošle se zpráva operačnímu systému, jaký soubor a s jakými parametry má systém přehrát.

Třetí modul obsahuje funkce pro analýzu signálů a je v něm definována struktura *TSIGNAL*. Tato struktura je základním stavebním prvkem programu obsahuje a veškeré informace o daném signálu. V této struktuře jsou uložena data signálu jeho délka, vzorkovací frekvence, příslušný druh spektra a všechny informace potřebné pro vykreslení signálu. Dále struktura obsahuje ukazatel na seznam složek, ze kterých je signál složen. Při jakékoliv změně složky je signál z těchto složek znovu vygenerován a je vypočteno i příslušné spektrum. Pro maximální univerzálnost programu jsou signály i složky uspořádány do lineárních spojových seznamů obr. 7.1. Seznam je navržen tak, že je v paměti uchován ukazatel na začátek seznamu a na aktuální prvek (tj. prvek, se kterým právě pracujeme). Každý prvek také obsahuje ukazatele na prvky, které se nachází v seznamu před a za ním. Pokud prvek nemá žádného předchůdce nebo následníka, je hodnota ukazatele nastavena na hodnotu *NULL*. Tato struktura je navržena tak, aby bylo kdykoliv možné jakýkoliv prvek ze seznamu odebrat a smazat nebo naopak přidat.



*Obr. 7.1 Lineární spojový seznam*

Posledním blokem programu je blok pro práci s jednotlivými složkami signálu. V tomto bloku jsou naprogramovány funkce pro vytvoření, resp. smazání složky nebo její změnu. Program umí pracovat s několika druhy složek. Složkou může být periodická funkce jako je sinus, nebo trojúhelníkový či obdélníkový signál. Dalším druhem může být šum, impuls, nebo zvuk wav. Posledním druhem složek je přímka. U této složky uživatel zadává, na jakém vzorku signálu přímka začíná a na jakém končí. A také zadává její počáteční a koncovou hodnotu.

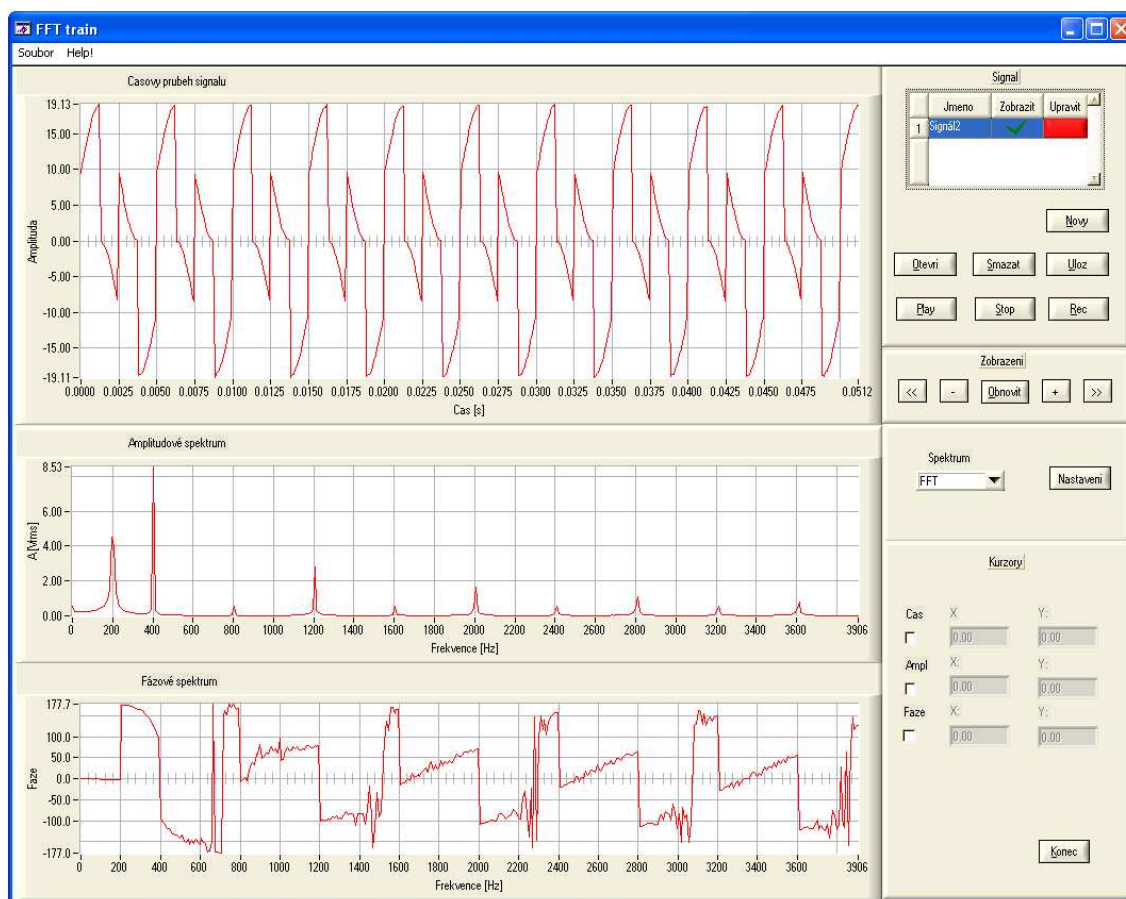
Výsledným složením kombinace různých složek může uživatel vytvořit téměř jakýkoliv průběh signálu.

## **7.2 Uživatelské rozhraní programu**

Uživatelské rozhraní programu tvoří tři základní panely. Hlavní panel zobrazuje časové průběhy signálů a jejich spektra. Na dalším panelu je generátor signálů, který umožňuje vytváření a úpravu složek signálů. Posledním základním panelem je panel s nastavením celé aplikace.

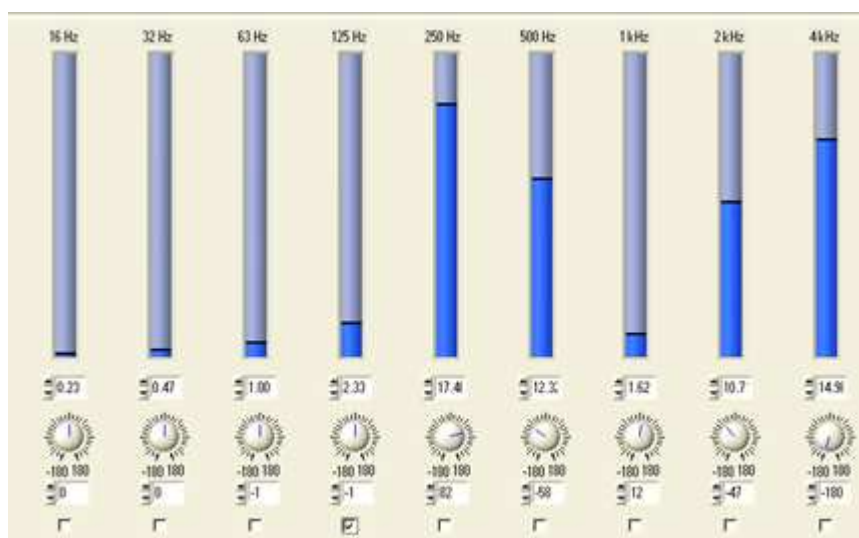
### **7.2.1 Hlavní panel aplikace**

Hlavní panel lze rozdělit do tří funkčních částí. První část tvoří graf, na kterém je zobrazen časový průběh signálu. Je-li signál složený z více složek, mohou se zde zobrazit i časové průběhy těchto složek. Zobrazený průběh signálu v grafu je možné různě přibližovat a posouvat jak pomocí ovládacích tlačítek, tak i tak pomocí myši v kombinaci s klávesou CTRL.



*Obr. 7.2 Hlavní okno programu*

Další částí hlavního panelu je část spekter. V této části je zobrazeno jedno ze čtyř druhů spekter, které program umí vypočítat. Základní volbou je zobrazení amplitudového a fázového spektra. Tato spektra jsou zobrazena ve dvou grafech a je samozřejmě možné spektrum v grafu přibližovat a posouvat pomocí myši. Další možností je zobrazení prvních patnácti složek z FFT. Hodnoty těchto složek nejsou zobrazeny v grafu, ale zobrazují je ovládací prvky, jako tomu je na obr. 7.3. Hodnotu, kterou tyto prvky zobrazují, je možné modifikovat pomocí tlačítek nebo myši. Z těchto hodnot je následně znovu vytvořen signál, který je zobrazen v grafu s časovým průběhem původního signálu. Tím má uživatel možnost vidět, jaký vliv má použití omezeného počtu složek na výsledný signál. Pod každým ovládacím prvkem je umístěno zaškrtnávací pole. Zaškrtnutím tohoto pole uživatel určí, zda se má daná složka použít při vytváření složeného signálu či nikoliv.



Obr. 7.3 Zobrazení amplitudy a fáze pomocí ovládacích prvků

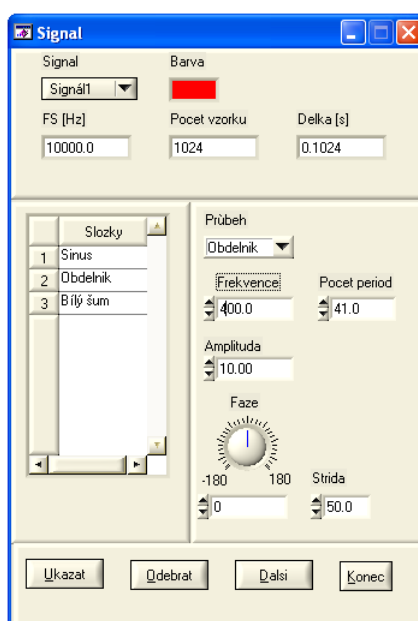
Dále program umí zobrazit oktávové a třetinooktávové spektrum. Tyto spektra jsou zobrazena stejným způsobem, jako jsou zobrazovány první složky z FFT. Takže ovládání a funkce těchto tří druhů spekter jsou prakticky stejné. Jsou zde jen nepatrné rozdíly. Například to, že první složka z FFT nemá fázi ani frekvenci, jedná se totiž o stejnosměrnou složku signálu. Dalším rozdílem jsou samozřejmě frekvence jednotlivých složek. Složky z FFT mají frekvenci v násobcích převrácené délky záznamu ( $1/T$ ), zatímco složky oktávového a třetinooktávového spektra mají frekvence odpovídající středním frekvencím jednotlivých pásem.

Poslední částí hlavního panelu je pruh s ovládacími prvky ležící v pravé části panelu. V horní části tohoto pruhu je tabulka signálů. V ní je uvedeno jméno každého signálu, jeho barva a to, jestli se má zobrazit. Dvojklikem na příslušnou buňku tabulky určíme, jestli se má signál zobrazit, resp. skrýt. Je-li zobrazeno amplitudové a fázové spektrum signálu, mohou být zobrazeny časové průběhy a spektra všech signálů v tabulce. Při volbě jiného druhu spektra, je zobrazeno spektrum signálu odpovídající aktuálnímu řádku v tabulce.

Pod tabulkou se nacházejí tlačítka pro vytvoření nového signálu, jeho smazání, uložení nebo přehrání. Po stisku tlačítka pro vytvoření nového signálu se objeví generátor signálů. Dále jsou zde tlačítka pro ovládání pohybu v grafu s časovým průběhem signálu. Ve spodní polovině panelu jsou pak zobrazovány hodnoty kurzorů jednotlivých grafů.

## 7.2.2 Panel generátor

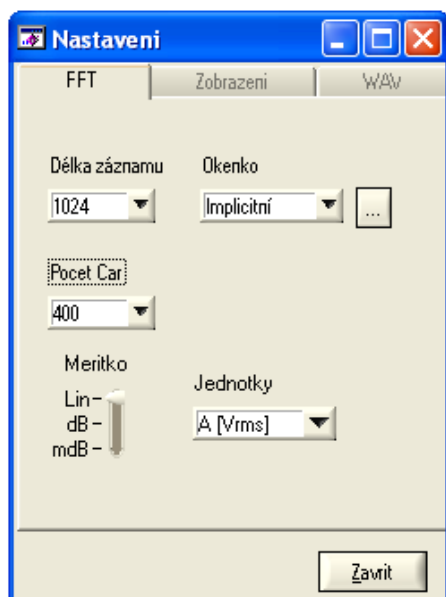
Tento panel umožňuje nastavení všech vlastností signálu, jako je jeho vzorkovací frekvence, délka a barva zobrazení. Délku signálu je možné zadávat jak pomocí počtu vzorků, tak i v sekundách. Signál sestavujeme postupným přidáváním jednotlivých složek. Tyto složky se zobrazují v tabulce a je kdykoliv možné je upravit nebo odstranit. Aby měl uživatel přehled o složce, kterou právě vytvořil, je zde tlačítko pro její zobrazení. Po jeho stisku se objeví vyskakovací okno s časovým průběhem této složky.



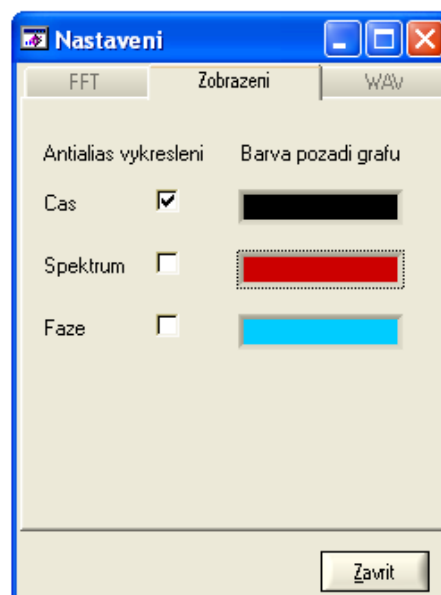
Obr. 7.4 Generátor signálů

## 7.2.3 Panel nastavení

Tento panel obsahuje různá nastavení programu, objeví se po stisku tlačítka „Nastavení“ na hlavním panelu. Je rozdělen do třech částí. První částí je nastavení FFT, volí se zde délka záznamu, počet zobrazených čar, typ okénkovací funkce případně její parametr. Můžeme vykreslit jeden signál s jednou okénkovací funkcí. Potom vykreslit další signál se stejnými parametry, ale s jinou okénkovací funkcí. Na grafu se spektrem signálů pak uvidíme rozdílné vlastnosti obou okénkovacích funkcí. Je zde i tlačítko pro zobrazení časového průběhu této funkce. Poslední dva ovládací prvky slouží k volbě měřítka a jednotek zobrazovaného spektra.



*Obr. 7.5 Nastavení FFT*



*Obr. 7.6 Nastavení zobrazení*

Další záložka obsahuje volby pro zobrazení signálu. Je možno nastavit barvu pozadí grafu a to jak je graf vykreslen. Volbou antialias kreslení je signál vykreslen v lepší kvalitě, ale vykreslení trvá podstatně déle. Poslední záložka obsahuje volby pro nastavení formátu výstupního wav souboru a pro nastavení kvality nahrávání.

## 7.2.4 Náповěda programu

Náповěda je dnes nedílnou součástí každého softwaru. Ke čtení rozsáhlé dokumentace uživatelům často chybí čas nebo chuť a právě pohotový systém nápovědy je pak vede při zadávání dat nebo spouštění různých úloh. Náповěda tak často plně nahradí různé příručky a další pomůcky.

Náповědu lze obecně realizovat několika základními prostředky: pomocí klasického souboru nápovědy WinHelp typu \*.hlp, pomocí HTML nápovědy (\*.chm) tedy kompilovaná HMTL nebo pomocí tipových oken (tooltip). Nabízejí se i méně rozšířené způsoby jako je použití PDF souboru se záložkami pro jednotlivé kapitoly nebo použití klasického \*.html souboru.

Použití systémové funkce WinHelp je od roku 1995 tradiční způsob nápovědy systému Windows. Vyžaduje připravit několikastránkový (často rozsáhlý) zdrojový dokument, přitom určitá místa označená speciální poznámkou pod čarou umožňují přímý přístup podle kontextu, podle klíčových slov, sekvenční prohlížení (browse) nebo textové hledání. Potřebné metody jsou součástí systému. Soubor může odkazovat na



další objekty, např. soubory s grafikou. Zdrojový soubor (může jich být i několik) je převeden do formátu RTF, obvykle exportem z vhodného textového procesoru. Překladač interpretuje pouze známé příkazy RTF, zbytek ignoruje. Hlavní zdrojový soubor HPJ má textový formát a na RTF soubory odkazuje. Překladem tohoto vznikne binární soubor HLP, který je zobrazován systémovým programem WinHelp.exe. Informace je zhuštěná, proto jeho rozsah není příliš velký ani při použití grafiky. Obsah nápovědy je obsažen v samostatném textovém souboru typu \*.cnt, který odkazuje na jednotlivé položky souboru \*.hlp, lze ho vytvářet libovolným textovým editorem.

Vytvoření nápovědy touto technikou vyžaduje níže popsanou úpravu programů. Informace pro nápovědu je v samostatném souboru, proto umožňuje přímou kontrolu jeho velikosti a v kritickém případě instalaci bez nápovědy. Zdrojové soubory jsou zpracovány v klasickém textovém procesoru, ale mají poněkud nezvyklou formu; potřebné systémové komponenty jsou společné pro všechny programy operačního systému a technika má plnou podporu ve vývojovém prostředí a knihovně MFC. Při využití malých okének se jeho využití přibližuje technice tipových oken. Pro vytváření vícejazyčných verzí je možno vytvářet soubory nápovědy individuálně pro různé jazyky.

Nápověda technikou HTMLHelp přišla na svět s Windows 97 a pro grafickou úpravu objektů využívá dnes běžný formát HTML používaný v internetu. Tato forma nápovědy je o něco více rozšířená než forma předchozí. Tento způsob nápovědy využívají zejména rozsáhlé aplikace, jako např. operační systém Windows, MS Office, apod. Oproti WinHelp má systém nápovědy HTML menší podporu knihovny MFC a vyžaduje přímé volání operačního systému. Umožňuje pružné rozšíření i na rozsáhlé systémy.

Hlavním parametrem pro volbu systému nápovědy byla kompatibilita se všemi dnes používanými verzemi operačního systému Windows. Tak aby byl program plně funkční na většině dnes používaných počítačů. Dalším kritériem byla možnost zakomponování do nápovědy i několik kapitol výukových textů. Tato kritéria zúžila výběr na volbu mezi klasickou nápovědou WinHelp a HTMLHelp. Nakonec byl vybrán systém nápovědy WinHelp. Který je v dnešní době asi nejrozšířenější a běžný uživatel je s ním obeznámen, což mu usnadňuje orientaci v programu. Nápověda byla vytvořena pomocí programu HTML Help WorkShop, který je zdarma k dispozici na stránkách společnosti Microsoft.

Samotný obsah nápovědy je rozdělen do dvou hlavních částí, první část nápovědy tvoří část výuková. V této části je popsán historický a teoretický úvod do teorie zpracování signálů a zejména pak Fourierovy transformace. Dále je zde uvedeno několik praktických informací týkající se praktického použití Fourierovy transformace, resp. analýzy s konstantní šířkou pásma.

Druhá část nápovědy je pak věnována samotnému programu, vysvětlení všech jeho funkcí a způsobu jeho ovládání.

### **7.3 Praktické ověření programu**

Ověření vlastností zvukové karty a její vhodnosti pro analýzu signálů je provedeno v kapitole 5.1. Správnost vypočtených hodnot FFT byla ověřena pomocí programu Matlab, kdy byly porovnány výsledky Fourierovy transformace vypočtené pro stejný signál. Hodnoty třetinooktávového a oktávového spektra byly porovnány s hodnotami vypočtenými programem Signal Analyzer, vytvořeným na VŠB TU Ostrava. Tyto vypočtené hodnoty se mírně liší. Odchyly závisí na druhu signálu. Rozdíl je způsoben použitím syntézy CPB z jednostupňové FFT, kde jsou spodní pásma dány součtem malého počtu frekvenčních čar. Jelikož je program určený pro výukové účely, je tento rozdíl nepodstatný.

## 8 Závěr

Výsledkem práce je program, který interaktivní formou seznámí uživatele s analýzou signálů pomocí rychlé Fourierovy transformace a analýzou s konstantní relativní šířkou pásma (CPB). Relativní šířka pásma je použita, poněvadž rozdělení spektra na třetinookávy lépe odpovídá subjektivnímu vjemu člověka. Program jednak ukazuje analýzu signálu rozkladem na jednotlivé harmonické složky, tak i syntézu (skládání jednotlivých složek) signálu. Pro získání časových průběhů signálů, které program analyzuje, obsahuje vlastní generátor. Signály je dále možné získat nahráním zvuku pomocí zvukové karty nebo načtením signálu ze souboru ve formátu wav. Analyzovaný signál je samozřejmě možné pomocí zvukové karty také přehrávat. Program neklade omezení na použití určitého typu zvukové karty, neboť lze-li danou zvukovou kartu nainstalovat pod operačním systémem Windows, pak jsou všechny tyto karty z hlediska softwarového přístupu shodné. Jsme tudíž pouze omezeni počtem bitů A/D resp. D/A převodníku a vzorkovací frekvencí měřící karty. Časový průběh signálu je možné modifikovat a sledovat, jak se tyto změny projeví ve frekvenčním spektru. A naopak zobrazené frekvenční spektrum jde rovněž upravovat změnou amplitud nebo fází jednotlivých složek a sledovat, jak se tyto změny projeví na časovém průběhu signálu. Program dále ukazuje některé další metody pro měření a vyhodnocování spekter signálů. Příkladem je použití různých okénkovacích funkcí a jejich vliv na spektrum signálu.

V nápovědě programu je stručné vysvětlení základů analýzy signálů a popis významu jednotlivých parametrů frekvenční analýzy. Nápověda obsahuje úvod do teorie Fourierovy transformace a její historii, stručný popis CPB analýzy a způsoby její syntézy. Tuto práci by bylo možné dále rozšířit do dalších oblastí analýzy signálů. Využitím více vstupů zvukové karty a vytvořit tím dvou a více kanálový analyzátor, na kterém by se dala předvést například vzájemná korelace dvou signálů.

## 9 Citovaná literatura

**Bruel & Kjaer. 1987.** *Technical Review No.3.* Naerum : Bruel & Kjaer, 1987. ISSN 007-2621.

**Hlaváč, Sedláček. 2007.** *Zpracování signálů a obrazu.* Praha : ČVUT, 2007. ISBN 978-80-01-03110-0.

**NI Spectral Leakage.** Spectral Leakage. *www.ni.com.* [Online] National Instruments. [Citace: 1. 2. 2007.] [http://zone.ni.com/reference/en-XX/help/371361B-01/ivanlsconcepts/spectral\\_leakage/](http://zone.ni.com/reference/en-XX/help/371361B-01/ivanlsconcepts/spectral_leakage/).

**NI The Fundamentals of FFT.** The Fundamentals of FFT-Based Signal Analysis and Measurement in LabVIEW and LabWindows/CVI. *NI Developer Zone.* [Online] [Citace: 6. 4. 2007.] <http://zone.ni.com/devzone/cda/tut/p/id/4278>.

**NI Third Octave Analysis Toolkit. 1995.** *Third Octave Analysis Toolkit.* Austin : National Instrument, 1995.

**NI Windowing Signals.** Windowing Signals. *www.ni.com.* [Online] [Citace: 2. 2. 2007.] [http://zone.ni.com/reference/en-XX/help/371361B-01/ivanlsconcepts/windowing\\_signals/](http://zone.ni.com/reference/en-XX/help/371361B-01/ivanlsconcepts/windowing_signals/).

**Nový, Richard. 2000.** *HLUK A CHVĚNÍ.* Praha : ČVUT, 2000. ISBN 80-01-02246-3.

**Oppenheim, Schafer. 1989.** *Discrete-Time signal processing.* Englewood Clifs : Prentice Hall, 1989. ISBN 0-13-216771-9.

**Petzold, Charles. 1998.** *Programming Windows.* Washington : Microsoft Press, 1998. ISBN 1-57231-995.

**Randall, R.B. 1987.** *Frequency analysis.* Naerum : Bruel & Kjaer, 1987. ISBN 87- 87355-07-8.

**Sonic.** Wave File Format. *Sonic Spot.* [Online] [Citace: 12. 4 2008.] <http://www.sonicspot.com/guide/wavefiles.html>.

**Tůma, Jiří. 1997.** *ZPRACOVÁNÍ SIGNÁLŮ ZISKANÝCH Z MECHANICKÁCH SYSTÉMŮ UŽITÍM FFT.* Praha : Sdělovací technika, 1997. ISBN 80-901936-1-7.

**Uhlíř, Sovka. 1995.** *Číslíkové zpracování signálů.* Praha : ČVUT, 1995. ISBN 80-01-01303-0.

**Wikipedia.org.** Joseph Fourier. *www.wikipedia.org.* [Online]  
[Citace: 9. 4. 2008.] [http://cs.wikipedia.org/wiki/Joseph\\_Fourier](http://cs.wikipedia.org/wiki/Joseph_Fourier).

**Wilson, Scott. 2003.** WAVE PCM soundfile format. *stanford.edu*.  
[Online] 20. 1 2003. Citace: 11. 04 2008]  
<http://ccrma.stanford.edu/courses/422/projects/WaveFormat/>.